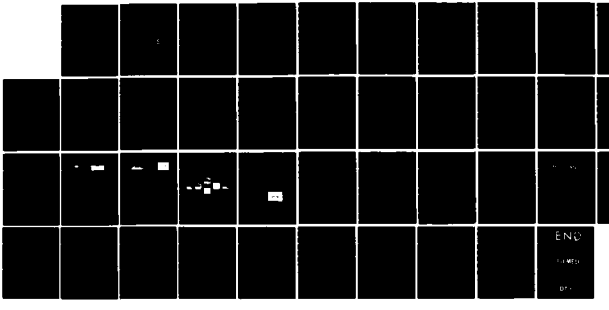
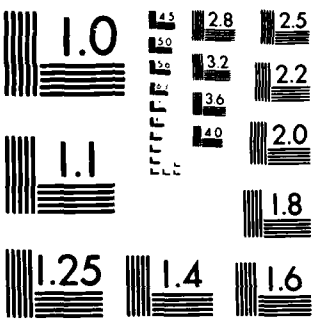
AD-A151 495 INTELLIGENT STATISTICAL TRACKER(U) RCA CORP CAMDEN N J 1/1
W B SCHAMING DEC 84 ARO-18485.3-EL DAAG29-81-C-0034

UNCLASSIFIED F/G 17/5 NL





UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ARO 18485.3-EL	2. GOVT ACCESSION NO.: N/A
3. RECIPIENT'S CATALOG NUMBER: N/A	
4. TITLE (and Subtitle): Intelligent Statistical Tracker	5. TYPE OF REPORT & PERIOD COVERED: 1 Sep 81 - 1 Oct 84 Final Report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): W. B. Schaming	8. CONTRACT OR GRANT NUMBER(s): DAAG29-81-C-0034
PERFORMING ORGANIZATION NAME AND ADDRESS: RCA Corporation, Camden, NJ 08485	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
CONTROLLING OFFICE NAME AND ADDRESS: U. S. Army Research Office, Post Office Box 12211, Research Triangle Park, NC 27709	12. REPORT DATE: December 1984
	13. NUMBER OF PAGES
MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DTIC ELECTE MAR 2 1 1985 S D B

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Target Acquisition
Target Tracking
Algorithms
Computerized Simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Advanced Technology Laboratories (ATL) of RCA has undertaken the task of developing sophisticated algorithms for autonomous target acquisition and tracking. When this contract was awarded, the investigators had already demonstrated in a

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

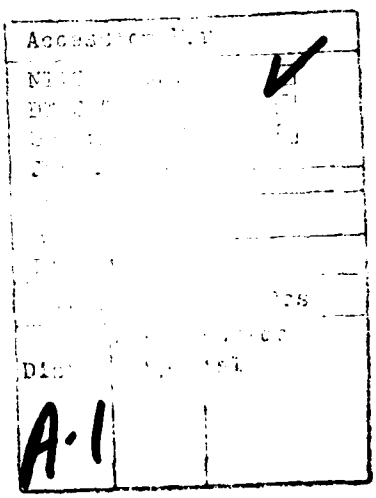
UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT CONTINUED:

simulation testbed that a statistical segmentation algorithm using multiple features for classification performs well on low resolution infrared images of tanks.

This contract was used to investigate improvements to the tracker system, and evolve towards a real-time hardware implementation.

The goals of the original contract were met within two years of the contract award. At that time, several analytical and experimental investigations had been conducted. The tracker simulator had gone through several revisions, as new concepts were invented and tested. A report describing many of the facets of the statistical tracking system that were worked out by this contract is attached as an appendix.



ARO 18485.3-EL

INTELLIGENT STATISTICAL TRACKER

FINAL REPORT

18 DECEMBER 1984

U. S. ARMY RESEARCH OFFICE

DAAG 29-81-C-0034

RCA CORPORATION

ADVANCED TECHNOLOGY LABORATORIES
MOORESTOWN CORPORATE CENTER, ROUTE 38
MOORSETOWN, NJ 08057

A. STATEMENT OF THE PROBLEM

The Advanced Technology Laboratories (ATL) of RCA has undertaken the task of developing sophisticated algorithms for autonomous target acquisition and tracking. When this contract was awarded, we had already demonstrated in a simulation testbed that a statistical segmentation algorithm using multiple features for classification performs well on low resolution infrared images of tanks. This work followed and extended the work of Dr. Gerald Flachs, Dr. Alton Gilbert, and others, working at White Sands Missile Range with Army Research Office support. We proposed with this contract to investigate improvements to the tracker system, and evolve towards a real-time hardware implementation.

B. SUMMARY OF IMPORTANT RESULTS

The goals of the original contract were met within two years of the contract award. At that time, several analytical and experimental investigations had been conducted. The tracker simulator had gone through several revisions, as new concepts were invented and tested. A report describing many of the facets of the statistical tracking system that were worked out by this contract is attached as an appendix.

The major accomplishments that occurred during the course of the contract were:

1. An adaptive gate process was implemented interior to the tracking window. The technique effectively controls the size of the tracking window.

2. The idea of intelligently defining varying weights for the misclassification costs C(T/B) and C(B/T) within

the tracking window was implemented and shown to improve tracking in multiple target situations. The technique was later dropped to allow for a new method of controlling the decision threshold.

3. Initial simulations revealed that a fixed threshold when computing the decision rule that segments out the target from the background was inadaquate. We observed that any fixed threshold would at some times grossly undersegment the target and at other times grossly oversegment the target. Given that there is a need for varying the decision threshold, two separate issues arise: what criteria to use to control the threshold, and how to control the threshold to satisfy the criteria. One of the accomplishments of the contract was the development of a satisfactory solution to both issues.

   The criteria by which the threshold is now controlled is essentially Neyman-Pearson, the probability of classifying a target pixel target (detection probability) is maximized under the constraint that the probability of misclassifying a background pixel (false alarm rate) is held to a fixed level. Additional constraints were added that serve to guarantee that at least some pixels will be labeled target.

   A computational procedure was developed that allows the threshold to be computed efficiently from the target and background histogram data. This development will allow the technique to be easily implemented in real-time.

4. A study was conducted of performance measures that might be used to sense and hopefully predict breacklock. Many potential performance measures were

defined and tested with both Gaussian and real image sequences. Several potentially useful statistical segmentation measures were found. The results of this investigation were then applied to the problem of assigning the number of bits to features.

5. A coast mode was developed and implemented in the simulator which significantly extended the tracking situations that can be successfully handled. The coast mode consists of three additions to the tracker simulator. First, a Kalman filter that maintains a running estimate of the target trajectory. Second, a performance monitor that is used to detect breaklock conditions. Last, a reacquistion strategy, that searchs for a target based on size, statistical signature, and trajectory.

6. In preparation for a proposal for a real-time, fieldable implementation of the tracker system, the simulator was used to finalize the version of the tracker to be proposed. At this time, this real-time system is being designed under contract with the U. S. Army Missile Command (MICOM). As a result of our efforts with the simulator, we were able to include many of the concepts described above (particularly breacklock detection and reacquisition) into the fieldable system.

7. An investigation was conducted on image representation and matching for motion analysis, which will have application to tracking and autonomous navigation. When applied to navigation, the motion occurs as the result of a moving sensor observing a stationary scene. The desired objective is to detect the motion of certain points in the scene, and analyze this motion to provide ranging or depth information and coarse object

segmentation. The formulated approach is to employ multiple processes on the images as follows: Laplacian pyramid, contour extraction and representation, structural matching, displacement field analysis, and object segmentation. Some preliminary results were obtained in the development of the contour extraction and representation process.

C. PUBLICATIONS AND TECHNICAL PRESENTATIONS

- o An Adaptive Gate Multifeature Baysian Statistical Tracker, 26'th International SPIE Convention, August, 1982.

- o Performance Measures for Statistical Segmentation, 26'th International SPIE Convention, August, 1982.

- o RCA Statistical Tracker, (Presentation) ARO Tracker Workshop, February, 1984.

- o Machine Vision for Motion Analysis (Presentation), IEEE Computer Society Workshop on Applied Imagery and Pattern Recognition, October, 1984.

D. PARTICIPATING SCIENTIFIC PERSONNEL

W. B. Schaming
J. S. Zapriala
R. Schellack
Dr. D. S. Shazeer
R. A. McClain (MS Systems Engineering, Univ. of Pa. 1984)
B. S. Mudge (MS Systems Engineering, Univ. of Pa. 1984)

Appendix: RCA's Statistical Tracking System

by R. A. McClain

RCA's Advanced Technology Laboratories
Camden, NJ

22 DEC 1983

Table of Contents

1.0 INTRODUCTION

This paper will describe the video tracking system developed by the Advanced Technology Laboratories of RCA Corporation on IR&D and under Army Research Office contract. A major theme is to explore the tradeoffs and interplay between the three major influences in the development of the tracker algorithm: theoretical pattern recognition, human intuition, and finally the constraint to be able to implement the tracker in real-time, that is, the tracker must be able to accept a standard video signal and process every video frame. The impact of this real time constraint will be highlighted throughout this paper.

The development of the tracker algorithm took place from 1981 to 1983 using a tracker system software simulator written in FORTRAN. The video inputs to the simulator were obtained by digitizing interesting sequences of video containing targets such as tanks, trucks, and airplanes. The video was generated by several types of sensors, both of the infrared and visible band. One of the important capabilities of this tracker is its ability to work with a variety of targets and sensors. The reason it can be flexible lies in the generality of the segmentation process, which does not rely on any a prior knowledge of the sensor or target type.

RCA has a contract with the U.S. Army Missile Command (MICOM) to build the real-time tracker system that so far has only been simulated.

2.0 OVERVIEW OF TRACKER ALGORITHM

A functional block diagram of the statistical tracker is shown in Figure 1. The major subsections of the diagram are the front end feature generator (video preprocessing, median filter, feature computation, scaling), the segmentation processing (histogram processing, decision rule computation, segmenter), high level control (centroid, adaptive gate window control, Kalman filter, mode control), and search processor.

The heart of the tracker is the statistical segmenter, which examines the image data and identifies those pixels that are target. The basic problem solved by the statistical segmenter is illustrated in Figure 2. Within the track window each pixel can be either target or background. Target pixels have one distribution of feature vectors (Figure 2 illustrates a single intensity feature) while background pixels most likely have some other distribution of feature vectors. Assuming the segmenter can learn what these distributions are, the problem is to use them to divide up the feature space into target and background regions. The statistical segmenter uses a likelihood ratio test to generate the target/background decisions. The complete process is described in a later section. In order to maintain good estimates of the feature distributions, the tracker must constantly update the window size and position in order to keep the target enclosed. The tracker incorporates a Kalman filter to estimate the velocity of the target. The velocity estimate modifies the position estimates from the last frame to establish a prediction window for the target location. This velocity prediction is especially important for targets with high line of site rates and in the reacquisition mode after breaklock has occurred.

To further increase the robustness of the tracker, a breaklock detection and target reacquisition capability has been developed. Breaklock is detected by examining the quality of the segmentation over a succession of frames. The target is reacquired by searching the binary target image for a target of appropriate size by correlation with a binary mask.

3.0 SUBSYSTEM DESCRIPTIONS

3.1 FEATURE GENERATION

The steps to generation of features are front end video preprocessing to digitize the video, median filtering that can be optionally bypassed, feature computation, and scaling/ combining.

The preprocessing consists of global automatic gain control, level correction and analog to digital conversion of the input video. The output is a two dimensional image, image(i,j), 60 times per second.

The median filter computes the 3 by 3 median of medians. Let the output of the median filter be denoted by pimage(i,j). To compute the value of the median at position i,j, we first compute the median (middle valued sample) of each of the three sets of intensities:

```
image(i-1,j-1), image(i-1,j), image(i-1,j+1)
image(i,j-1),   image(i,j),   image(i,j+1)
image(i+1,j-1), image(i+1,j), image(i+1,j+1)
```

We then compute the median of these three median values. The primary benefit of median filtering is for noise reduction when tracking targets in noisy video.

There are a number of features that can be used in the tracker, and we have run tests using various combinations. We have found that two features, intensity and edge magnitude, are generally optimal.

The edge feature is an approximation to the Sobel edge magnitude. The calculation is shown in Figure 3. The intensity feature is the sampled, filtered video intensity. Scaling of features is necessary because the segmenter uses, at most, a total of 8 bits from both features combined. Typically, we use 4 bits of intensity and 4 bits of edge, but any combination that uses no more than 8 bits total will be operator selectable. The limitation to eight bits is partly implementation driven, but also driven by the need to be able to estimate joint probability density functions of the features. If too many bits are allowed then the amount of data needed to make the estimates accurately becomes prohibitive. We have typically used a total of 7 or 8 bits in the simulations, which has worked well even for small targets (which generate less data to estimate the probability densities).

The scaler scales the feature samples by any power of two (saturating the number if it exceeds the range of legal numbers), and then selects the specified number of bits from the the most significant end of the number.

To aid the processor in setting the power of two scale factor, the scaler measures the peak value of each feature within the track window every frame.

3.2 SEGMENTATION

The tracker uses a target segmentation technique based on optimum decision theory. It is a widely applicable technique in that the decision rule is derived totally from the video with the only assumption being that the target is initially designated to the tracker. No assumption is implicit that the target is the brightest (or hottest) object in the field of view. The tracker can, of course, track such targets, because it quickly learns that the target is the brightest object.

The derivation of the segmentation decision rule will be described in several steps. First, the hypothesis test is stated. Several solutions to the hypothesis test are possible, depending on what criteria is used to make the decisions. The criteria selected for this tracker is similar to the Neyman-Pearson criteria, so it is described first. Then the modified segmentation criteria used by the tracker to derive the decision rule is described. Finally, the methods by which the theory is actually applied in the tracker are discussed.

Hypothesis Test - The formulation of the sementation process begins with a description of the segmenters job in the form of a hypothesis test: The segmenter is given samples of the feature vector, X, at each pixel position. The segmenter must classify the pixel as either target or background. One of the two choices must be made. The assumption is now made that the conditional probability densities are known:

p(X¶T) - Probability of observing X, given that the underlying pixel is target.

p(X¶B) - Probability of observing X, given that the underlying pixel is background.

These densities will be estimated by the segmenter with histograms.

Decision Criteria - Several criteria for making the decision are possible. In the past, we have used a Bayes risk criteria to making the decision. With this approach, one assigns costs to the classification mistakes (decide target when actually the pixel is part of the background, or vice versa). More recently, we have settled on a criteria that is similar to the Neyman-Pearson criteria. The concern is with the probability of false alarm, Pf, which is the probability of deciding target when the underlying pixel is background, and

the probability of detection, Pd, which is the probability of deciding target when the underlying pixel actually is target. With the Neyman-Pearson criteria, one constrains Pf#<=#k, and designs a test to maximize Pd, while satisfying the constraint.

The optimum solution to the Neyman-Pearson hypothesis test is based on the likelihood ratio (for derivation see Van Trees, Detection, Estimation, and Modulation Theory, Part I, pg. 33-34.):

$$\frac{p(X\P T)}{p(X\P B)} \underset{\text{Background}}{\overset{\text{Target}}{\gtrless}} \text{Lambda}$$

The threshold Lambda is set so that Pf=k.

One complication occurs with the whole derivation, due to the fact that what is actually estimated in the segmenter is not p(X¶T) and p(X¶B), but instead the distributions within the window and frame regions: p(X¶W) and p(X¶F). In order to apply the above optimum test, the following assumption is made: That the target lies completely within the window region, leaving only background in the frame. With this assumption, the distributions within the window can be expressed in terms of the target and background distributions, as follows:

$$p(X\P W) = A*p(X\P B) + (1-A)*p(X\P T)$$
$$p(X\P F) = p(X\P B)$$

These equations say that the window distribution is given by a weighted sum of the target and background distributions, while the frame distribution is equal to the background distribution, because by assumption, no target is allowed in the frame.

The pair of equations can be solved for the window and frame distributions, which can then be substituted into the hypothesis test to give:

$$\frac{p(X \mid W)}{p(X \mid B)} \underset{\text{Background}}{\overset{\text{Target}}{\gtrless}} \text{Lambda}'$$

where Lambda' = Lambda*(1-A)+A. The constant A is the fraction of the window that is background. The value of A is not known precisely, but it is not needed anyway, because the technique which is used to set the threshold, directly sets Lambda'. Henceforth, the ' will be dropped and Lambda' will simply be called Lambda.

The above discussion has placed on firm ground the fact that the tracker uses window and frame distributions to generate the decision rule. Because the segmenter uses a likelihood ratio test, for a given false alarm rate, the probability of detection is maximized. Now the method by which the threshold, Lambda, is set will be described.

As Lambda is decreased, the resulting decision rule will classify an increasing number of features as target. To set Lambda, we apply two rules. First, we require that at least some minimum fraction of pixels within the window be classified target. So we automatically lower Lambda until that fraction is reached. Second, we lower the value of Lambda further until some maximum fraction of pixels in the frame region are classified target. This second rule maximizes the fraction of pixels in the window region that will be classified target.

The effect of the first rule is that even in situations which are very difficult to segment, such as high clutter situations, the segmenter will at least classify some minimum amount of the window region as target. The effect of the second rule is that, in easy to segment situations, the segmenter will be able to pull out the whole target. As the difficulty of the scene increases the segmenter will back off and classify only the most likely pixels as target. In simulations, we have settled on a value of .15 for the minimum fraction of target in the window. Generally, the window control algorithm will drive the window size to an area such that the target fills between .25 to .30 of the window. So only in very high clutter situations will the segmenter be forced to exceed the desired false alarm rate in the frame region. The maximum fraction of target pixels in the frame region has been set to. 015 in our simulations. The application of this theoretical discussion of decision theory by the tracker will now be described.

Every frame, the segmenter collects histograms of the features within the window and frame regions. The window histogram h(X¶W) indicates the number of times the feature combination X occurred within the window region, and the frame histogram h(X¶F) indicates the number of times the feature combination X occurred within the frame region. These histograms are normalized by the number of pixels in each of the regions to give estimates of the probability distributions in the window and frame regions. These instantaneous (single frame) distributions will be denoted pi(X¶W) and pi(X¶F).

In order to reduce short term fluctuations and improve the quality of the estimates, the instantaneous distributions are filtered in time over several video frames to obtain smoothed distributions, denoted ps(X¶W) and ps(X¶F). The filtering operation is given by:

$$ps(X¶W)[new] = C * ps(X¶W)[old] + (1-C) * pi(X¶ W)$$
$$ps(X¶F)[new] = C * ps(X¶F)[old] + (1-C) * pi(X¶F)$$

The constant C is chosen to provide 10 to 15 frames of averaging, with exponential weighting.

The likelihood ratio is computed for all X:

$$L(X) = \frac{ps(X\P W)}{ps(X\P F)}$$

In order to set the threshold, we need to be able to determine for any value of Lambda, how many pixels will be classified target in both the window and frame regions. The functions of interest will be denoted N(Lambda¶W) and N(Lambda¶F). They are defined by:

N(Lambda¶W) = Sum h(X¶W) over all X satisfying L(X) >= Lambda

N(Lambda¶F) = Sum h(X¶F) over all X satisfying L(X) >= Lambda

It might first appear that some sort of iterative search for the threshold is necessary. Such an approach might not be feasible considering the fact that this computation must be performed 60 times/second in the real-time version of the tracker. The following somewhat novel approach was conceived to allow the computation to be non-iterative, with the sacrifice of accuracy in the final value of Lambda, because Lambda will be forced to one of a finite set of values.

We can compute the N(Lambda¶W) and N(Lambda¶F) functions at discrete values of Lambda = k*D with a two step process. First, we compute auxilliary functions:

dN(k¶W) = N(k*D¶W) - N((k+1)*D¶W)
dN(k¶F) = N(k*D¶F) - N((k+1)*D¶F)

These functions are sort of like a histogram in the threshold. They determine for a given value of the threshold, what change in the numbers of target pixels in the window and frame region would be caused by decreasing Lambda by the amount D. The dN functions are computed with the following algorithm:

For all k:

dN(k¶W) = 0
dN(k¶F) = 0

For all X:

z = L(X)/D

k = Truncate z to integer and in the range [1..256]

dN(k¶W) = dN(k¶W) + h(X¶W)
dN(k¶F) = dN(k¶F) + h(X¶F)

The functions N(Lambda¶W) and N(Lambda¶F) can now be evaluated at the discrete points Lambda = k*D by

N(k*D¶W) = Sum dN(i¶W) over all i from k to 256
N(k*D¶F) = Sum dN(i¶F) over all i from k to 256

In practice these two functions are computed starting at the maximum k (256) and then decreasing k until first the number of target pixels in the window region, N(k*D¶W), exceeds Pd times the area of the window. Then k is possibly further decreased, until the number of target pixels in the frame region, N(k*D¶F) would exceed Pf times the area of the frame region. The final value of k is then used to compute the value of Lambda using Lambda = k*D

The decision rule to be used during frame i is computed during frame i-1 using smoothed statistics up to frame i-2 and using the instantaneous histograms from frame i-2 to set the threshold.

The decision rule is stored in the segmenter in the form of a look up table that, for each feature combination, contains the binary decision. Incoming feature combinations, X, are classified by looking up the decision. The output of the segmenter is a binary image consisting of the one or zero decisions for each pixel.

3.3 TARGET IMAGE PROCESSING

Target image processing entails many processes. The target image is a binary image that represents the segmenters decisions as to which pixels are target. The target image is processed to obtain the target position using projections, and the target size using an adaptive gate process. These estimations, which are made every frame, are processed by a high level controller in order to predict the size and position of the target in subsequent frames. The window is placed in each frame in the position where the target is anticipated to be on that frame. The next section will describe the additional processing on the target image for breaklock detection and reaquistion.

For the computation of the target position, first, projections are computed within the target window. The row projection is obtained by summing the number of target pixels in each row across the columns. The column projection is obtained by summing the number of target pixels in each column

down the rows. These two projection functions are then processed to determine the 50% points, that is the row and column which split the area under the projection functions into two equal halves.

The use of 50% points is preferred over a true centroid. A true centroid (center of gravity) weights points that are farther from the center more heavily than points that are closer in to the center.

The position of the target is processed through a Kalman filter of order two (in each of horizontal and vertical directions), which maintains a velocity and position estimate of the target. The low order of the Kalman filter is due partially to the difficulty of modeling the target motion in the sequences that we have been using in the simulator. In the sequences we use, sensor motion is also responsible for motion of the target in the scene. At some later point in time, when we actually close the loop and the tracker drives the sensor pointing angle, we will have to revisit the issue of target modeling. In any case the position estimate is used each frame to position the target window where the target is predicted to be according to the Kalman state extrapolation. Also, the Kalman state vector is used during the reacquistion mode to bias the search for the target to where the target would most likely be located.

The dynamic control of the window size is accomplished using an adaptive gate process which provides a means of increasing or decreasing the size of the window as the target size changes due to target motion or range closure. Edge gates are placed in positions which are expected to lie on target boundaries. Counts of target pixels are taken within these gates. The size of the window is controlled by a servo which drives the window size so that the number of target pixels balances the number of background pixels in the gates. The

imbalance in any frame is passed through a low bandwidth digital filter to generate the new window size. The filter includes two cascaded integrators, therefore the closed loop is a type two servo. It can respond to an expanding target size with zero residual error.

3.4 BREAKLOCK DETECTION AND REACQUISTION

The breaklock detection and reacquistion system is provided as a last resort to prevent loss of track. In simulations of the tracker system, targets have been reacquired after temporarily disappearing completely behind an occlusion. The system should be able to handle reacquistion after loss of track due to sensor jitter at launch.

Complete execution of breaklock detection and reacquisition occurs in three phases, corresponding to the three tracker modes: normal track, track while coast, and search. For the detection of breaklock, a segmentation performance measure is monitored. The performance measure is an estimate of the probability of misclassifying a pixel in the target window. It is computed by counting the number of target pixels in the window and frame regions. When the tracker is in normal track mode, the performance is monitored. If the performance measure falls below a certain threshold in any given frame, then the mode is switched to the track while coast mode. Whenever the transition to the coast while track mode occurs, several key target parameters are saved: the histograms for computing the decision rule, the target size, and the target trajectory (in the form of the Kalman state vector estimate). While in the track while coast mode, tracking continues normally, but now a running average of the performance measure is computed. After at least six frames in this mode, the average performance measure is compared to two thresholds. If it rises above the first threshold, then the assumption is made that the cause of

performance degradation was temporary and therefore ignorable, thus the tracker returns to normal track. If on the other hand, the performance measure falls below the second threshold then the assumption is made that breaklock has occurred. In this case the tracker enters the search mode to try to reacquire the target.

The search for the target is made within the binary target image. The decision rule used to generate the target image is computed using the histograms that were saved at the time of initial performance degradation. The binary mask that is correlated with the target image is shown in Figure 4. The size of the binary mask is set according to the size of the target. The mask shape and size are such that the correlation will peak when at a blob of target pixels of about the correct size. Successful reacquisition of the target is detected by comparing the peak correlation to a threshold. The final piece of information that is used to help find the target is the known target trajectory. The trajectory is used to generate a penalty function that is subtracted from the spatial correlation function. The effect of the penalty function is to bias the search towards the predicted target position and to limit the region searched. The search region is steadily increased for every frame that the target goes undetected. Eventually (after a couple of seconds) the search region encompasses the complete field of view.

4.0 SUMMARY

This paper has attempted to describe the tracking system that has been developed by RCA. Many of the techniques used were a result of the contraint to be able to process video in real-time. The use of histograms to estimate the probability densities involved is one example. The simple performance measure used to detect breaklock is another.

Also, at times the only justification for a design decision is human intuition or experience derived through the simulator. An example of such a design tradeoff was the decision to use a modified Neyman-Pearson criteria that guaratees at least a minimum probability of detection no matter what the false alarm rate.

A Reprint from the

PROCEEDINGS

Of SPIE-The International Society for Optical Engineering

Volume 359

Applications of Digital Image Processing IV

August 24-27, 1982
San Diego, California

Adaptive gate multifeature Bayesian statistical tracker

W. B. Schaming
RCA Advanced Technology Laboratories
Camden, New Jersey 08102

Adaptive gate multifeature Bayesian statistical tracker

W. B. Schaming
RCA Advanced Technology Laboratories
Camden, New Jersey 08102

Abstract

A statistically based tracking algorithm is described which utilizes a powerful segmentation algorithm. Multiple features such as intensity, edge magnitude, and spatial frequency are combined to form a joint probability distribution to characterize a region containing a target and its immediate surround. These distributions are integrated over time to provide a stable estimate of the target region and background statistics. A Bayesian decision rule is implemented using these distributions to classify individual pixels as target or nontarget. An adaptive gate process is used to estimate desired changes in the tracking window size.

Introduction

This paper documents progress during the past year toward the development and demonstrations of a statistical tracking algorithm. Papers[1,2] presented in 1981 described some of the initial concepts in this development. Since that time, the statistical tracking algorithm has been expanded to incorporate (a) the simultaneous use of multiple features, (b) an adaptive gate process for control of the window size, and (c) positional dependence of the misclassification cost factor.

The tracking algorithm is based on the use of multifeature joint probability density functions for the statistical separation of targets from their background. The features currently being used are intensity, edge magnitude, and a pseudo spatial frequency feature. These features are combined to form the joint distributions which characterize a target region and its immediate surround. The distributions are integrated over time to provide a stable estimate of the target and background statistics. A Bayesian decision rule is implemented using these distributions to classify individual pixels as target or nontarget within a tracking window. An adaptive gate process is used to estimate desired changes in the tracking window size. The algorithm at present assumes manual target designation.

RCA believes this tracking process is capable of operation in all environments, insensitive to target type, signature, and orientation; applicable to a variety of sensors; and extendable to multisensor processing and readily implementable.

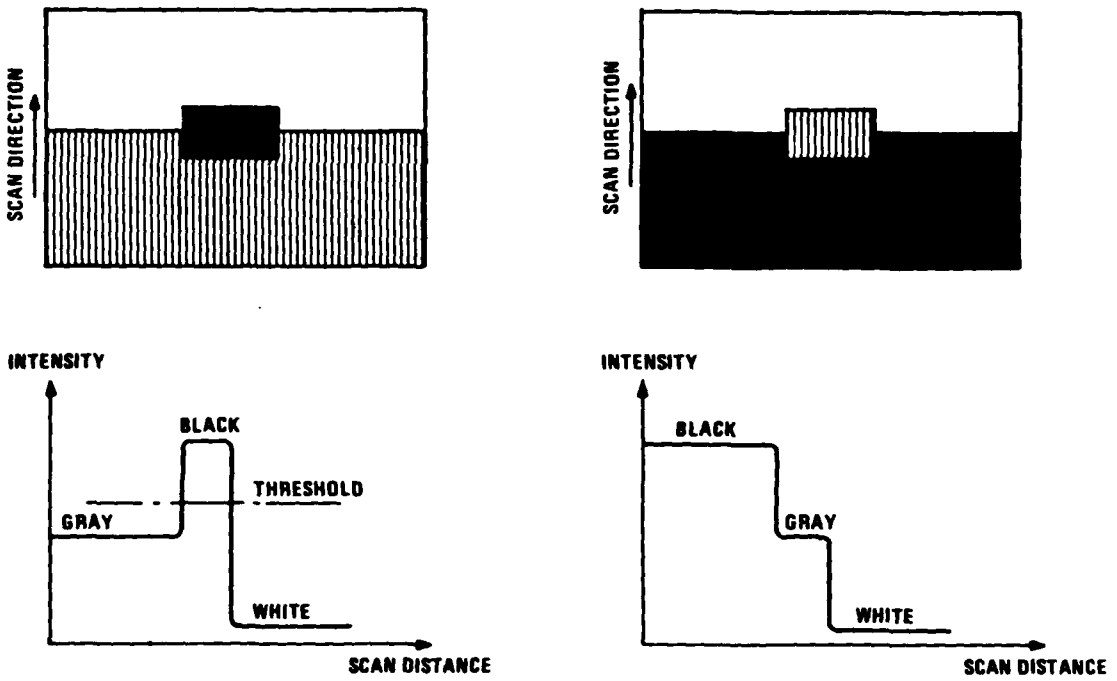
Preprocessing and A/D conversion

The video preprocessing function is an important part of any imaging sensor system, but is more critical when the sensor is an IR device which may exhibit very high dynamic range capability. In this case it is insufficient to perform a simple AGC based upon global statistics because the subsequent rescaling to reduce the dynamic range will destroy the low contrast local detail. Instead, some form of local adaptive contrast enhancement should be applied in which the gain varies with the local contrast. Lo[3] simulated and compared several such techniques.

Although necessary in a hardware implementation, this function has not been included in the simulations reported here. Ten-second image sequences were digitized from video tape via an analog video disc and an image processing system. The input to the image processing system was passed through a video processing amplifier so that the levels could be properly matched to the A/D converter.

Statistical tracking algorithm

Targets are often separated from their background by a simple thresholding scheme. Sometimes the computation of the threshold is quite sophisticated and involves looking at the statistics of the video signal. However, thresholding is inherently limited in ability as can be seen by the diagrams in Fig. 1. A simple black and white target can be readily thresholded to isolate it from its background. On the other hand a gray target cannot be thresholded without using a pair of thresholds properly placed to contain the intensity levels on the target. This dual threshold in itself is not prohibitive, but rather the problem lies in the ability to place the thresholds at the appropriate levels.



(a) THIS TARGET CAN BE EASILY THRESHOLDED

(b) THIS TARGET CANNOT BE EASILY THRESHOLDED BUT REQUIRES A PAIR OF THRESHOLDS PROPERLY PLACED.

Fig. 1. Example showing two postulated targets. One is easily segmented from the background using a single threshold. The other, however, requires two thresholds which are not easily determined. The statistical process provides a separate threshold for each intensity level.

The statistical segmentation process is a technique which provides an improved method for extracting the target from its background. Figure 2 depicts this process. Shown are two histograms, one taken from a window area of the image containing the target and the other taken from the immediate surround which represents the background. A single feature, intensity, is shown in these histograms for illustrative purposes. The shape of the distribution shown is arbitrary; there are no assumptions made about their actual shape. The segmentation process makes a separate assessment of each bin in the histogram to determine if pixels whose intensity falls in the bin are more likely to be target or background. In addition to solving the threshold selection problem, the statistical tracking algorithm provides a method to both simplify the multimode tracking concept and provide added capability.

The simplification comes about in the following way. State-of-the-art multimode trackers typically operate a contrast, edge, and correlation tracker in parallel. An executive process may be defined to determine at any given time which tracking mode is providing the most reliable estimate of target position. The statistical process, as currently defined, eliminates this mode polling process by combining the available features into multidimensional statistics representing target and background. Consider the use of intensity and edge magnitude as the two candidate features. In this case the statistical approach encompasses three tracking modes in an integrated single mode without the need to poll the performance of the individual processes. When intensity is the best target background separator, the algorithm operates like a contrast tracker. When edge magnitude is predominate it operates similar to an edge centroid tracker. Because the process is searching for pixels in the current frame that are statistically similar to those pixels selected as target in previous frames, the algorithm is in a sense a correlation type process as well.

The added capability comes from the fact that there are target/background conditions which are inseparable using two features independently but are readily separable using the same two features jointly. This is illustrated quite simply in Fig. 3. In this example, neither edge magnitude nor intensity can be used independently to separate the target from background because both flat distributions cover the entire variable range for both features. On the other hand, the joint distribution clearly delineates the two areas.

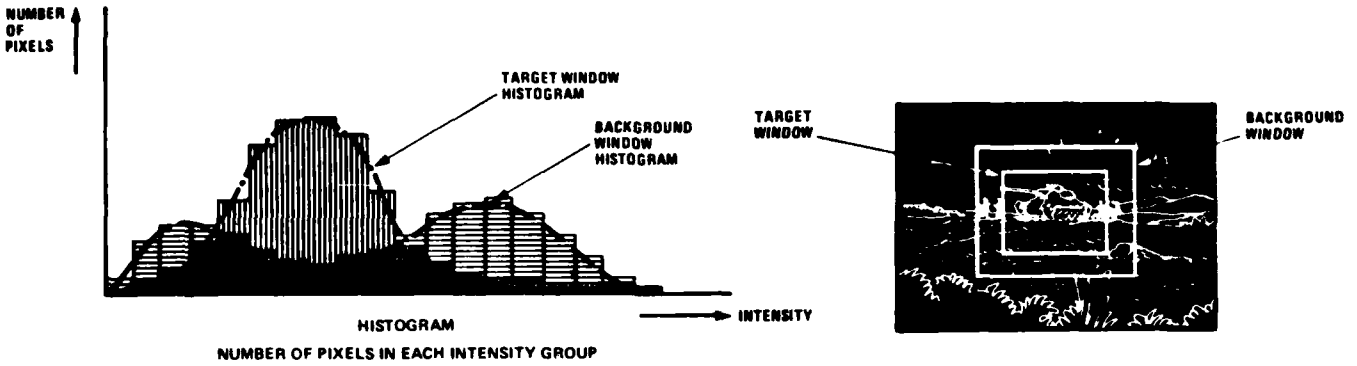


Fig. 2. Example of how histograms are used to separate a target from its background. Each bin in the histogram is examined to determine if the intensity value falling within that bin are more likely to be target or background. Although this is a single feature (intensity) example, the same process is used with multiple features in an N-dimensional histogram representing a joint probability density.

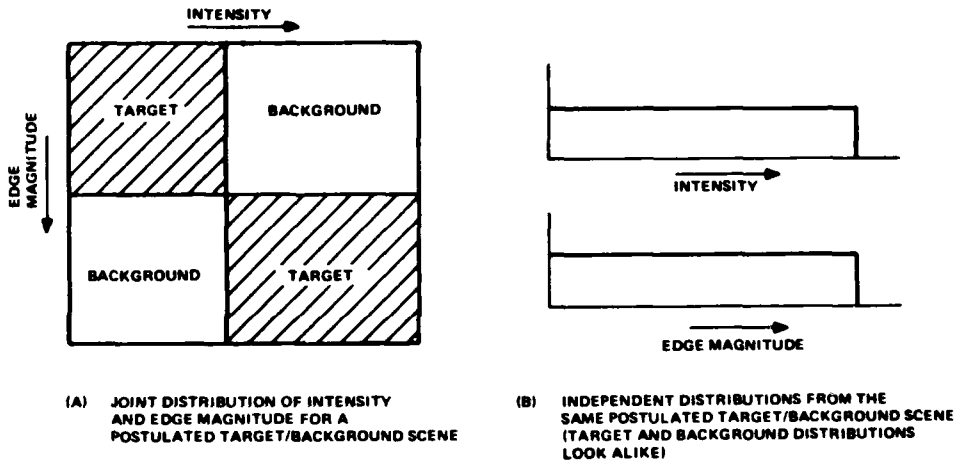


Fig. 3. Simple example showing how the use of joint statistics aids in the separation of target from background in situations where the use of the features singly fails.

Figure 4 is a flow diagram of the statistical tracking mode. The preprocessed video is used to generate multiple feature images to be used in the decision process. The features are combined into two joint probability density functions for (a) a target tracking window and (b) a background window frame. These distributions are the basis of a statistical decision process which is used to classify the image pixels inside the tracking window to separate the target from the background. In actuality the statistics from previous frames are used in the classification process for the current frame. At the same time, histograms are generated from the current image frame so that the statistics can be updated for processing subsequent frames. At the end of the classification process the segmented image is analyzed to determine the appropriate error signals as well as the window size and position for the next frame. In parallel with the pixel rate computations for the Nth frame, the statistics from the N-1st frame are integrated with past history and a decision rule is generated for the N+1st frame.

A sample output from the process is shown in Fig. 5. Only two features were used for this example, namely, intensity and edge magnitude. The total number of bits utilized for the features is seven — four for intensity and three for edge magnitude. The edge magnitude used is the absolute value approximation to the Sobel operator.

The next few paragraphs describe some of the steps in this process in more detail.

Computation of features

The first step in the statistical process is the generation of the features to be used. There are many potential candidates, some of which are computationally too burdensome for real-time implementation at this time. We therefore have limited our selection of features

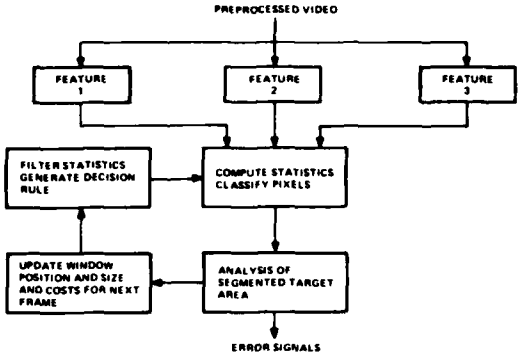


Fig. 4. Block diagram of the Bayesian statistical tracking mode. The feature computation, statistics generation, and pixel classification are performed at the pixel rate. The computation of error signals is performed during vertical sync.

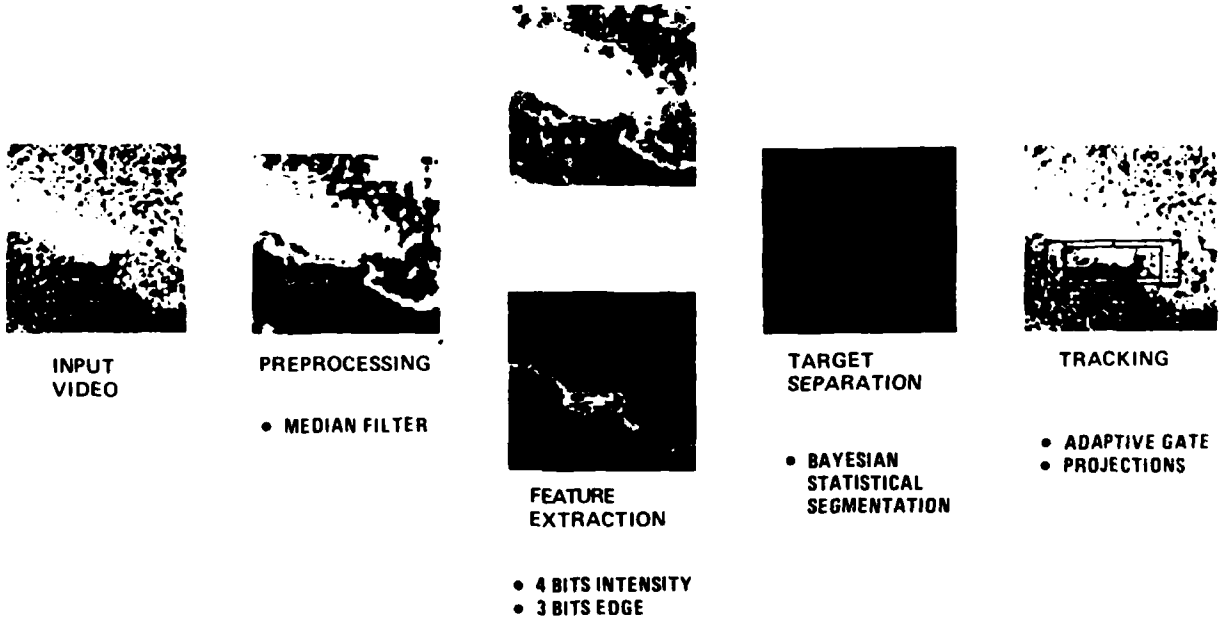


Fig. 5. Sample output from the Bayesian statistical tracker simulation using a 64-x-64 pixel image of an aircraft at a mountain boundary. Two features were used in the statistical segmentation with a total of seven bits.

to those which are readily implemented. These features are intensity, edge magnitude, and spatial frequency.

The intensity feature is simply a requantized version of the digitized video signal to obtain the desired number of bits of intensity resolution. The edge magnitude feature is the sum of absolute values approximation to the Sobel operator. The absolute sum is an acceptable and computationally more appealing approximation than the true edge magnitude.

The third feature is an approximation to spatial frequency in the horizontal direction. Because it is a measure of object size, it could also be considered a simple texture measure in a broad sense. The spatial frequency is defined as the function of the run length where a run is the number of consecutive pixels between which the pixel-to-pixel difference does not exceed a predefined threshold. The threshold used is the mean value of the absolute difference beweeen pixels in the previous frame. The feature value is then defined as:

$$SF = \text{MAXIMUM} \left[0, \; (2^N - \text{RUN LENGTH}) \right] \tag{1}$$

where 2^N is the number of levels into which the spatial frequency feature will be quantized.

An example of the spatial frequency feature is shown in Fig. 6. An arbitrary function is plotted to represent the image intensity I at successive pixels in the x direction. Beneath the plotted data are shown the actual pixel intensities, absolute differences, run lengths, and feature values. The threshold used to compute run lengths in the example is 1.3 and the number of quantization levels is 8(N = 3 bits). The first sample-to-sample difference which exceeds the threshold 1.3 is the sixth sample. Samples 1 to 5 represent a run of length 5 in which the differences do not exceed threshold. The corresponding feature value is 3 which is assigned to all pixel locations in the run. The higher feature values indicate smaller distances between gradient values exceeding threshold. Note that the low amplitude variation between the pixels 6 and 14 do not exceed the threshold and therefore do not define the boundary of a run. The feature is intended to provide information about the size (in the x direction) of areas or patches which have uniform or slowly varying intensity.

Generation and integration of statistics

Histograms from two separate regions in the image must be computed to provide the probability density functions required by the decision rule. The regions from which the histograms are generated are shown in Fig. 7. The assumption in the segmentation algorithm is that the target is absent from the frame region. For both the frame and window regions a multifeature histogram is defined as

$H^N_{FR}\ (f_1, f_2, f_3)$ Frame Region Histogram

$H^N_{WR}\ (f_1, f_2, f_3)$ Window Region Histogram

for the Nth image in the sequence.

After normalization by the respective areas of the frame and window regions the histograms become the discrete joint probability densities

$P^N_{FR}\ (f_1, f_2, f_3)$

$P^N_{WR}\ (f_1, f_2, f_3)$.

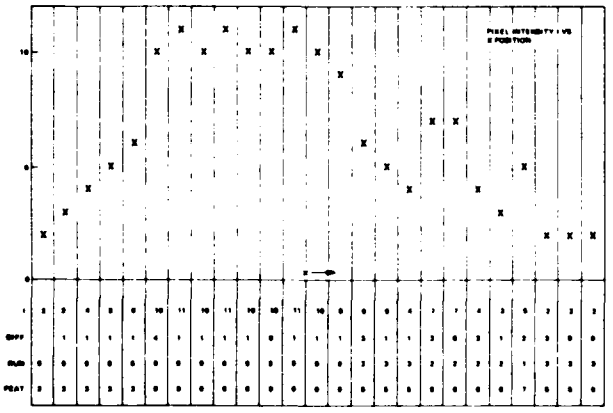


Fig. 6. Sample which shows the procedure for calculating the pseudo spatial frequency feature. The absolute difference threshold used to compute run lengths in the example is 1.3, which is the average difference. The number of quantization levels for the feature is 8.

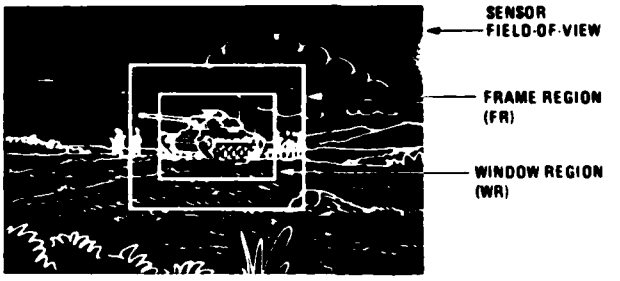


Fig. 7. Areas of the image over which the multifeature histograms are computed. It is assumed that the target is absent from the frame region which is defined as a border around the window region containing the target.

To minimize short-term statistical variations these probability densities are combined in a weighted sum with the past history of the statistics. This fading memory filtering is performed once each frame so that the statistical updating keeps up with the frame rate of the video. The filtering is defined by

$$FP_{FR}^{N} = a\ P_{FR}^{N} + (1 - a)\ FP_{FR}^{N-1} \tag{2}$$

$$FP_{WR}^{N} = b\ P_{WR}^{N} + (1 - b)\ FP_{WR}^{N-1} \tag{3}$$

where

FP_{FR}^{N}, FP_{WR}^{N} are the filtered probability density functions at the Nth frame time

P_{FR}^{N}, P_{WR}^{N} are the unfiltered density functions computed from the current frame N

a, b are the weighting factors which control the amount of smoothing performed.

In the simulations performed to date, the filtered statistics up to and including frame N-1 are used to generate the decision rule to be used on frame N.

Minimal cost decision rule

The decision rule used in the classification of pixels as target or background inside the window region is based upon minimizing the cost or risk associated with making a particular choice. A pixel is called a target pixel if, and only if, the cost (or penalty) associated with deciding background is greater than the cost associated with deciding target. Mathematically this is written as follows:

Decide target if and only if

$$P(B/\bar{X})C(B/B) + P(T/\bar{X})C(B/T) > P(B/\bar{X})C(T/B) + P(T/\bar{X})C(T/T) \tag{4}$$

where

$P(B/\bar{X})$ is the probability of a pixel being background given that the pixel has the feature vector $\bar{X}$.

$P(T/\bar{X})$ is the probability of a pixel being target given that the pixel has the feature vector $\bar{X}$.

C(B/B) is the cost associated with classifying a background pixel as background.

C(T/T) is the cost associated with classifying a target pixel as target.

C(B/T) is the cost associated with classifying a target pixel as background.

C(T/B) is the cost associated with classifying a background pixel as target.

Clearly C(B/B) and C(T/T) are zero because there should be no penalty for making a correct decision. The decision rule then becomes the following:

Decide target if and only if

$$P(T/\bar{X})C(B/T) > P(B/\bar{X})C(T/B). \tag{5}$$

Using Bayes theorem this inequality is rewritten as

$$P(\bar{X}/T)P(T)C(B/T) > P(\bar{X}/B)P(B)C(T/B). \tag{6}$$

These distributions can be expressed in terms of the window and frame regions shown in Fig. 7. Because the basic assumption is that the target is absent from the frame region, the background distribution is the same as the frame region distribution, therefore,

$$P(\bar{X}/FR) = P(\bar{X}/B). \tag{7}$$

The window region contains both target and background, therefore

$$P(\bar{X}/WR) = P(B)P(\bar{X}/B) + P(T)P(\bar{X}/T) \tag{8}$$

where

$P(B)$, $P(T)$ = the a priori probability of background and target respectively in the window region.

Substitution of $P(\bar{X}/B)$ and $P(\bar{X}/T)$ results in

$$P(\bar{X}/WR) > \left[\frac{C(T/B) + C(B/T)}{C(B/T)}\right] P(B) P(\bar{X}/FR) \tag{9}$$

which is a decision rule based upon the measurable distributions in the frame and window regions. This is more simply written as

$$P(\bar{X}/WR) > \alpha A \; P(\bar{X}/FR) \tag{10}$$

where

$$\alpha = P(B)$$

$$A = \frac{C(T/B) + C(B/T)}{C(B/T)}$$

The parameters A and α play an important role in the overall process. In the operation of the tracking algorithm, α is maintained approximately constant by attempting to maintain a fixed relationship between the size of the target and the window size. In subsequent paragraphs it will be shown how the parameter A is used both as a control parameter as well as a means of introducing pixel position into the decision process. The parameter A is referred to as the misclassification cost.

Window size and position control

The dynamic control of the window size is accomplished using one of the most successful techniques of modern trackers, namely the adaptive gate process. This provides a means of increasing or decreasing the size of the window as the target size changes due to target motion or range closure. The approach being used places the appropriate edge gates inside the statistical tracking window as shown in Fig. 8. The central area defined by the heavy black lines is the area in which the segmented target is confined. For ideal operation the edge gates would contain half background and half target pixels. The adaptive gate process is an attempt to balance the number of target and background pixels in the horizontal and vertical edge gates independently to control the height and width of the window. The unbalance in the edge gates is the difference between the number of target pixels and background pixels. To control the horizontal window size this unbalance is used to either expand or contract the window size horizontally. If ΔW_E as defined in Fig. 8 is positive, there is more target area than background area in the edge gate regions. The gates are then expanded. Conversely, if ΔW_E is negative there is more background area which suggests that the gates should contract. The effect is to make a change in the width of the central region in which the target is being contained.

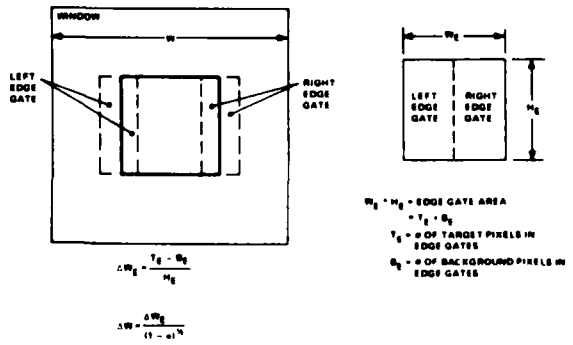


Fig. 8. Target edge gates are located inside the window area. The change ΔW_E in the combined width of the two-edge gates is related to the background vs. target area unbalance within the gates. This change then defines the change ΔW to be made in the tracking window width W to maintain a constant nominal value for α. A similar process defines the change ΔH in the window height.

Recall in the previous paragraph it was stated that α (the a priori probability of background in the window region) could be maintained approximately constant. This is accomplished during the window size control process. If ΔW_E is the desired change in width of the central target area as derived from the edge gates, then the corresponding change in the statistical tracking window width W is defined as

$$\Delta W = \frac{\Delta W_E}{(1-\alpha)^{1/2}} \tag{11}$$

A similar process defines the change in the window height. This process tries to maintain a central target area which is $(1-\alpha)$ times the window area.

The probability of background inside the window is then approximated by α.

The relative position of the tracking window with respect to the image frame is controlled in one of two ways. In the case of bench test video tape input (or in fact any input which is not directly controlled by the error signals) the error signals must drive the position of the tracking window for the next frame. On the other hand when the tracking error signals are driving a seeker, the window position within the frame will vary only in certain circumstances such as the initiation of a search and reacquisition strategy.

The error signals are computed as the difference between the centroid of the segmented target region and the current position of the window in the frame.

Misclassification cost control

The misclassification cost, A, is controlled in two ways. First, it is a function of the pixel position relative to the expected aim point and second, it is adjusted in a control loop using the parameter α as a reference. This in effect puts a positional dependency into the decision rule so that the classification of a pixel as a target point is a function of its location in the image relative to the current best estimate of target location. The adaptive gate mechanism provides an ideal method for assigning relative weights to the two cost factors C(T/B) and C(B/T). Figure 9 shows the layout of the frame and window regions along with a plot in the X direction of the relative magnitude of the composite cost function A. The area labeled R1 in the figure should correspond to the central area of the target assuming the adaptive gate window control function is performing properly. In this region we expect mostly target pixels. The penalty for mislabeling a background pixel as target in area R1 therefore should be less than the penalty for calling a target pixel background. In the edge gate regions R2 there should be half background and half target pixels in the ideal case; therefore the misclassification costs C(T/B)and C(B/T) should be equal. In the area R3 between the edge gates and the window boundary very little target area is anticipated. This provides a buffer zone between the target and the frame region which is assumed to contain no target data. Consequently, the relative magnitudes of the misclassification costs should reverse. Finally, because the frame region should not contain any target pixels, the penalty for misclassifying background pixels as target in this region should be even higher.

A similar function is applied in the Y direction and the actual misclassification factor is the larger of the two. This does not, however, set the actual magnitude of A which is required for given image conditions. Consequently the overall amplitude of the parameter A is made adaptive. This is done in the following way.

The two parameters in the decision rule which impact the pixel classification in addition to the statistics are the misclassification cost A and the a priori probability of background in the tracking window α. It is desirable to hold α at a constant value inside the window. Therefore, $(1-\alpha)$ is used as the reference parameter in a simple control loop as shown in Fig. 10. After classifying the pixels in an image frame, the adaptive gate

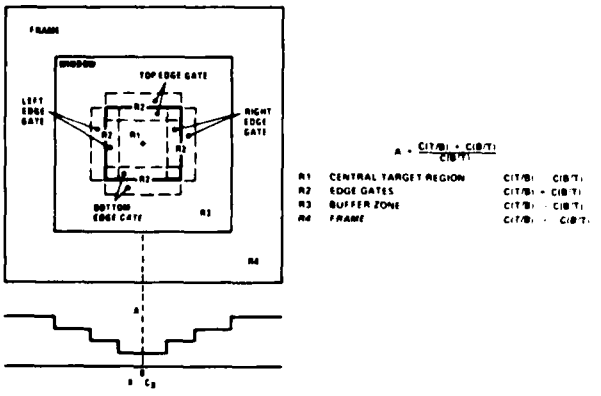


Fig. 9. Diagram showing the positional dependence of the misclassification cost. A plot of the relative amplitude of "A" as a function of the pixel position is shown for the X direction.

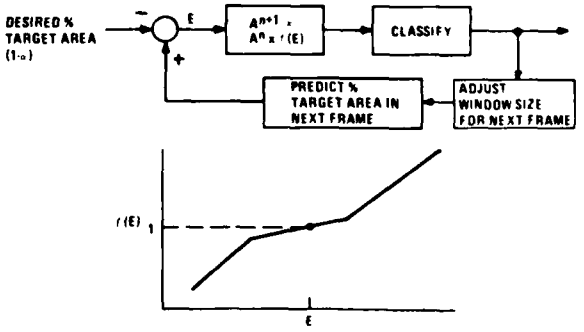


Fig. 10. Control loop used to adjust the misclassification cost used in frame n (A^n) to a new value to be used for frame n+1 (A^{n+1}).

process is used to set the window size for the next frame. The number of pixels expected to be classified as target in the next frame is estimated to be the same as the current frame. Using the estimated target size and the window size calculated for the next frame, the percentage target area expected in the next frame can be computed. The predicted target area is compared to the desired reference ($1-\alpha$) to obtain an error which defines a scale factor by which the cost function A is scaled. The adjustment in A will tend to improve the classification in the next frame. Lowering the magnitude of A will cause more pixels to be classified as target and vice versa.

Conclusions

A statistical tracking algorithm has been demonstrated via simulation which incorporates the concepts of a multimode tracker in a single mode. The use of multifeature joint probability distributions provide better target separation than using the same features individually. The statistical process is insensitive to sensor type and operating scenario which provides a wide range of applicability without the need for application dependent training. The technique is also insensitive to target orientation (such as that caused by platform roll) because no specific target-related information is assumed. The algorithm is directly extendable to multisensor operation which would provide a wider range of operating conditions.

Acknowledgment

The development of the algorithms reported here have been supported in part by the U.S. Army Research Office under contract DAAG29-81-C-0034. The author expresses his appreciation to Dr. G.M. Flachs for many enlightening discussions on this subject and to J. Hayman for suggesting the spatial frequency feature.

References

1 W.B. Schaming, R.C. Skevington, and G.M. Flachs, "Real-Time Statistical Tracker for Infrared (IR) Focal Plane Array," Proceedings of SPIE, Vol. 302, pp. 178-1985, August 1981.

2 R.C. Skevington, G.M. Flachs, and W.B. Schaming, "A Statistical Approach to Image Segmentation," Proceedings of PRIP 81, pp. 267-272, August 1981.

3 C.M. Lo, "A Survey of FLIR Image Enhancement," Proceedings of 5th International Conference on Pattern Recognition, Vol. 2, pp. 920-924, December 1980.

A Reprint from the

PROCEEDINGS

Of SPIE-The International Society for Optical Engineering

Volume 359

Applications of Digital Image Processing IV

August 24-27, 1982
San Diego, California

Performance measures for statistical segmentation

Dov J. Shazeer
RCA Advanced Technology Laboratories, Camden, New Jersey 08102

Performance measures for statistical segmentation

Dov J. Shazeer
RCA Advanced Technology Laboratories, Camden, New Jersey 08102

Abstract

Performance measures for statistical segmentation have been developed for a space-and-time critical Bayesian statistical tracker. They are intended to become an integral part of a knowledge-based tracking algorithm, which has been developed by RCA. The performance measures are serving to quantify the usefulness of the processed input, to assist in the identification of each tracking state and give its reliability, and to predict impending changes of state. They have been tested using stochastically generated target-background frames. Performance measure results have correlated well with the parameters which characterize the difference in the target and background distributions. A host of possible performance measures are discussed in relation to their strengths and weaknesses. Experimental results for the measures currently being employed by RCA are given, and areas for future research are indicated.

Introduction

Space-and-time constrained tracking algorithms have suffered from a lack of intelligence. Recent work on a statistical approach to a 2-dimensional image segmentation using Bayesian decision criteria has given hope that this deficiency can be remedied.[1,2] During the past two years the RCA Advanced Technology Laboratories has had considerable success in the development of a Multifeature Bayesian Intelligent Tracker (MFBIT).[3] An intelligent tracker must adapt to rapidly changing tracking conditions. MFBIT is intended to handle a variety of tracking strategies which are dependent on the tracking conditions. In addition, it must identify tracking conditions and impending changes in them. A knowledge-based tracking processor is being implemented which responds to the current input based upon a time-series record of measures extracted from previous inputs. The tracking conditions are implemented as a finite state automaton. The anticipated tracking states are (1) target acquisition, (2) multiple targets, (3) breaklock, (4) target leaving field-of-view, and (5) target confusion. The performance measures have been developed to add to the intelligence of MFBIT. They will be utilized (1) to predict and identify changes of tracking state, (2) as an experimental instrument to define each tracking state, (3) in a near-optimal allocation scheme of computational resources to the competing features, and (4) in the determination of Kalman weights by which the present input is integrated with the past to reduce wild oscillations in tracking strategy.

Picture segmentation

Picture segmentation is a procedure which locates the target and background domains of an image. Each image is initially digitized, to form a 2-dimensional array of pixels. Using appropriate operators, features such as intensity, edge magnitude, texture, etc., can be extracted. Based on its feature values, each pixel is assigned to the states-of-nature "target" or "background." A Bayesian decision rule determines the set of feature values J_T and J_B which causes the pixels to be assigned the state-of-nature "target" or background," respectively.

The Bayesian decision rule is obtained by making the assumption that the set of pixels belonging to the state-of-nature "target" will have a different statistical distribution of feature values than pixels belonging to the state-of-nature "background." Two regions are therefore defined for each frame in a sequence of images: a rectangular window region (WR) containing pixels which belong to both states-of-nature, enclosed in a frame region (FR) whose pixels have the states-of-nature, "background" only (see Fig. 1). We obtain the probability density histograms, $h^{FR}(J)$, from the frame region. $h^{FR}(J)$ corresponds to the conditional probability, P(J/BR), that if the pixel is from the background region, then the chance variable will have the feature value J. The window region probability density histogram, $h^{WR}(J)$, is obtained. This probability, P(J/WR), that the chance variable will have a feature value J in the window region, is the expectation value, summed over both states-of-nature, of the conditional probability of feature value J.

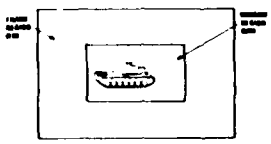


Fig. 1. The window region contains both target and background pixels. The frame region contains background pixels only.

Because only the window region is segmented, P(J/WR) is simply written as P(J). Then,

$$P(J) = P(T)P(J/T)+P(B)P(J/B) \tag{1}$$

where P(T) is the a priori probability of the state-of-nature "target" in the WR and P(B) is the a priori probability of the state-of-nature "background" in the WR. For the sake of simplifying notation, P(B) α, P(T) $1-\alpha$, P(J/T) $h^T(J)$, and P(J/B) $h^B(J)$. If we now assume that $h^B(J) = h^{FR}(J)$, and that we can reasonably estimate α then equation 1 can be used to solve for $h^T(J)$

$$h^T(J) = \frac{1}{1-\alpha}\left\{h^{WR}(J) - \alpha h^{FR}(J)\right\} \tag{2}$$

Bayes theorem allows a calculation of a posteriori probabilities from a knowledge of the a priori probabilities.

$$P(T/J) = \frac{P(T)P(J/T)}{P(J)} = \frac{(1-\alpha)h^T(J)}{h^{WR}(J)} \tag{3}$$

and

$$P(B/J) = \frac{P(B)P(J/B)}{P(J)} = \frac{\alpha h^{FR}(J)}{h^{WR}(J)} \tag{4}$$

The a posterior probabilities are used in decision rules. Several are extant. The rule that a pixel whose feature value is J will be labeled "target" if

$$P(T/J) > P(B/J) \tag{5}$$

otherwise it will be labeled "background" is known as the maximum likelihood rule. Using equations 3 and 4 we may restate the rule. Label the pixel "target" if

$$\frac{h^T(J)}{h^{FR}(J)} \quad \frac{\alpha}{1-\alpha} \tag{6}$$

In the 1920s it was shown by Neyman and Pearson[4] that optimum decision rules are formulated in terms of the ratio $(P(J/T)/P(J/B)$. This ratio, $h^T(J)/h^{FR}(J)$ for the tracker, is known as the likelihood ratio. The value it must exceed (i.e. $\alpha/1-\alpha$) is called the decision criterion. The maximum likelihood rule guarantees that the majority of the pixels are correctly labeled.

Another Bayesian decision rule is known as the minimum risk rule. The risk in labeling the pixels "target" or "background" is calculated by defining the misclassification costs C(B/T) and C(T/B). C(B/T) is the cost of labeling "background" a pixel whose state-of-nature is "target," and C(T/B) is the cost of labeling "target" a pixel whose state-of-nature is "background." The Bayesian risk in labeling "target" a pixel whose feature value is J is R(T/J,B) = C(T/B) P(B) P(J/B). Likewise, the Bayesian risk in labeling a pixel whose feature is J as "background" is R(B/J, T) = C(B/T) P(T) P(J/T). The Bayesian decision rule is to label a pixel to minimize the risk. This leads to the rule: a pixel whose feature value is J will be labeled "target" if

$$C(T/B)P(B)P(J/B) < C(B/T)P(T)P(J/T) \tag{7}$$

otherwise, label the pixel "background."

Equation 7 demands that a pixel be labeled "background" unless

$$\frac{h^T(J)}{h^{FR}(J)} > \left[\frac{\alpha}{1-\alpha}\right]\left[\frac{C(T/B)}{C(B/T)}\right] \tag{8}$$

We can rewrite equation 8 by using equation 2.

$$\frac{h^{WR}(J)}{h^{FR}(J)} > \left[\frac{C(B/T) + C(T/B)}{C(B/T)}\right] \alpha \qquad (9)$$

$$\text{define } A \equiv \frac{C(B/T) + C(T/B)}{C(B/T)}$$

Thus, the decision criterion for a Bayesian classifier is $A\alpha$.

It is of interest to note that the minimum risk and maximum likelihood rules are identical for equal misclassification costs (i.e., $C(T/B) = C(B/T)$).

Estimation of performance

The Bayesian decision rules partition the set of feature values $\{J\}$ into two mutually exclusive sets, $\{J_T\}$ and $\{J_B\}$. If a pixel in the window region has a feature value which belongs to the set $\{J_T\}$, it will be labeled "target," regardless of its state-of-nature. In a like manner, if the pixel has a feature value belonging to the set $\{J_B\}$, it will be labeled "background." Because the histograms $h^T(J)$ and $h^{FR}(J)$ usually overlap, this labeling scheme will lead to the misclassification of some pixels. The magnitude of the error, i.e., the number of pixels misclassified, will depend on the amount of overlap and on the value of the decision criterion, $A\alpha$. The number of misclassified pixels is a minimum if $A = 2$, the maximum likelihood decision rule. However, other values of A may minimize, in some fashion, the risk of misclassification. In any case, our choice of the decision criterion reflects our bias. Thus, performance measures which take the partitioning of the set $\{J\}$ into account will be called biased performance measures. Performance measures which compare the shapes or functional forms of the histograms, $h^T(J)$ and $h^{FR}(J)$, will be called unbiased performance measures. Both types of estimations are useful.

Biased performance measures

Hit rate and false alarm rate

Assume a decision rule divides observation space into two disjoint sets, $\{J_T\}$ and $\{J_B\}$, as in Fig. 2.

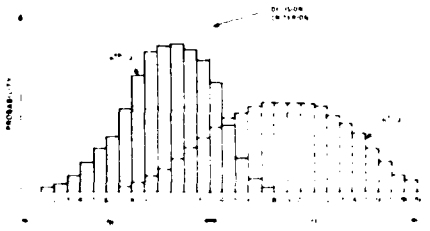


Fig. 2. Probability distribution histograms.

It can be seen that $\sum_{\{J_T\}} h^{FR}(j) = P(T/B)$ is the probability that a pixel state-of-nature is "background" is labeled "target." Alternatively, $P(B/T) = \sum_{\{J_B\}} h^T(j)$ is the probability that a pixel whose state-of-nature is "target" is labeled "background." $P(T/T) = \sum_{\{J_T\}} h^T(J) = 1 - P(B/T)$ is the probability that a pixel whose state-of-nature is "target" will be labeled "target." $P(T/T)$ is called the hit rate (HR). $P(B/T)$ is the miss rate (MR). $P(T/B)$ is the false alarm rate (FAR), and $P(B/B)$ is called the correct rejection rate (CRR). Because we locate the centroid of the target by operating in some manner on the pixels labeled "target," it is evident that a false alarm is generally more costly than a miss. Note that HR and FAR cannot be independently varied. From Fig. 2 it is seen that they depend on each other implicitly through the decision criterion.

Bayes risk, error

We can calculate the total risk of mislabeling pixels by calculating the risk of mislabeling background pixels,

$$R(T/B) = \sum_{\{J_T\}} R(T/J,B) = \alpha C(T/B) \quad \text{(FAR)} \qquad (10)$$

and adding to it the risk of mislabeling target pixels

$$R(B/T) = \sum_{\{J_B\}} R(B/J,T) = (1-\alpha)\, C(B/T)\, (1-HR) \qquad (11)$$

The sum is known as Bayes risk:

$$\text{Bayes} = R(T/B) + R(B/T) = \alpha C(T/B)(FAR) + (1-\alpha)\, C(B/T)(MR) \tag{12}$$

The total number of misclassified pixels will be called ERROR:

$$\text{ERROR} = N\,[\alpha\ (FAR) + (1-\alpha)(MR)] \tag{13}$$

where N = total number of pixels in the window region.

ERROR is minimized for the maximum likelihood conditions C(T/B) = C(B/T). The ERROR = N times the shaded area under the graph (see Fig. 3). If the decision criterion is moved to the right, the decrease in false alarms is smaller than the increase in the misses. If the decision criterion is moved to the left, the decrease in misses is smaller than the increase in false alarms.

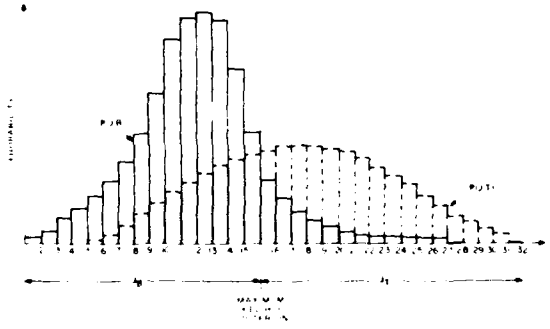


Fig. 3. Probability density distribution.

Another performance measure which is a function of the HR and FAR is the ratio of densities (ROD):

$$\text{ROD} = \frac{HR}{FAR} = \frac{\sum_{J_T} h^T(J)}{\sum_{J_T} h^{FR}(J)} = \frac{P(T/T)}{P(T/B)} \tag{14}$$

Because $h^T(J) > \left[\frac{\alpha}{1-\alpha}\right]\left[\frac{C(T/B)}{C(B/T)}\right] h^{FR}(J)$ if $J \in \{J_T\}$

$$\therefore \sum_{\{J_T\}} h^T(J) > \left[\frac{\alpha}{1-\alpha}\right]\left[\frac{C(T/B)}{C(B/T)}\right] \sum_{J_T} h^{FR}(J)$$

$$\therefore \text{ROD} > \left[\frac{\alpha}{1-\alpha}\right]\left[\frac{C(T/B)}{C(B/T)}\right]$$

If ROD is large relative to $\left(\frac{\alpha}{1-\alpha}\right)\left[\frac{C(T/B)}{C(B/T)}\right]$, the Bayesian classifier is performing well.

It would obviously be helpful in the accurate location of the target, if the number of false alarms were a small fraction of the total number of pixels labeled "target." Therefore, a performance measure called the false alarm fraction (FAF) has been defined:

$$\text{FAF} = \frac{\text{Fraction of false alarms}}{\text{Fraction of pixels labeled target}} = \frac{\alpha(FAR)}{\alpha(FAR+(1-\alpha)HR)} = \frac{\alpha \sum_{J_T} h^{FR}(J)}{\sum_{J_T} h^{WR}(J)} \tag{15}$$

In Fig. 3, if we move the decision criterion to the right one bin, the total number of errors increase, but the FAF decreases. We probably will locate the target centroid more accurately.

Unbiased performance measures

These measures compare the histograms $h^T(J)$ and $h^{FR}(J)$, independently of the choice of the decision criterion. However, if these unbiased measures show that $h^T(J)$ and $h^{FR}(J)$ are extremely similar in functional form, no amount of cleverness will allow us to segment the image. On the other hand, if the unbiased measures show the histograms to be different, but the biased performance measures show that segmentation is poor, we have the opportunity to remediate. Thus, the unbiased performance measures can be used to monitor the performance of the biased performance measures. The two types of measures should agree.

Cos θ

A simply implemented yet effective measure called cos θ can be utilized. Suppose the chance variable J is partitioned into N grey levels. We may view $h^T(J)$ and $h^{FR}(J)$ as two, N-dimensional, generalized vectors, $\vec{h}^T$ $\vec{h}^{FR}$. Their dot product would be $\vec{h}^T \cdot \vec{h}^{FR}(J) = |\vec{h}^T| |\vec{h}^{FR}| \cos\theta = \sum_J h^T(J)\, h^{FR}(J)$ solving for cos θ we have

$$\cos\theta = \frac{\sum_{J=1}^{N} h^T(J)\, h^{FR}(J)}{\left\{ \sum_{K=1}^{N} \left[h^T(K)\right]^2 \sum_{L=1}^{N} \left[h^{FR}(L)\right]^2 \right\}^{\frac{1}{2}}} \tag{16}$$

Schwartz's inequality assures us that the denominator is always less than or equal to the numerator. Note that if $h^T(J) = c \cdot h^{FR}(J)$ for all J then cos θ = 1 and θ = 0°. If, on the other hand, $h^T(J) = 0$, if $h^{FR}(J) \neq 0$ or $h^{FR}(J) = 0$, if $h^T(J) \neq 0$, then cos θ = 0° and θ = 90°. In the first case either all the pixels will be labeled "background" or they will all be labeled "target."

Unreliability parameter

The unreliability parameter (UPAR) measures the probability that a target pixel is likely to be labeled background. We define $U_p(J) = P(B/J)\, P(J/T)$ as the probability that a pixel whose state-of-nature is target and whose feature value is J is labeled "background." Summing $U_p(J)$ over all J results in the unreliability parameter.

$$U_p = P(B/T) = \sum_J U_p(J) = \sum_J P(B/J)\, P(J/T)$$

However, $P(J/T) = h^T(J)$

and $$P(B/J) = \frac{\alpha h^{FR}(J)}{h^{WR}(J)}$$

Therefore, $$U_p = \alpha \sum_J \frac{h^T(J)\, h^{FR}(J)}{h^{WR}(J)} \tag{17}$$

There are two extremums. First, $h^T(J) = h^{FR}(J)$ for all J, and in this case, $U_p = \alpha$. Second, there is no overlap between $h^T(J)$ and $h^{FR}(J)$, i.e., $h^T(J) = 0$ whenever $h^{FR}(J) \neq 0$ and $h^{FR}(J) = 0$ whenever $h^T(J) \neq 0$, and in this case $U_p = 0$. It will take on intermediate values for intermediate cases of overlap. However, if α approaches either one or zero, $U_p \to \alpha$. Therefore, it is wise to scale U_p by α. Values of U_p close to zero indicate favorable conditions for segmentation, while values close to one indicate poor segmentation conditions.

Because $U_p = P(B/T)$, it is interesting to note that the $P(T/B) = (1-\alpha)\frac{\sum_J h^T(J)\, h^{FR}(J)}{h^{WR}(J)}$

$= \frac{(1-\alpha)}{\alpha} U_p$ and thus there is a symmetry between the two performance measures.

Weighted second moment

The Weighted Second Moment (WSM) is derived by rewriting equation 1 in the following manner:

$$h^{WR}(J) = \alpha\, h^{FR}(J) + (1-\alpha) h^T(J)$$

If we define $$y = \frac{h^{WR}(J)}{h^{FR}(j)} \quad \text{and} \quad x = \frac{h^T(J)}{h^{FR}(J)} \tag{18}$$

we have $y = \alpha + (1-\alpha)\, x$

Equation 18 is the equation of a straight line with intercept α and slope (1-α). The line goes through the point (1,1). The set of data points [x(J), y(J)] can be distributed in any fashion on the line. In fact the distribution of the points on the line is used as a performance measure. The point (1,1) has special significance. At that point, $h^T(J) = h^{WR}(J) = h^{FR}(J)$, and only the crudest type of image segmentation can occur. If the set of points {(x,y)} are confined to a small interval about the point (1,1), segmentation is poor. If, on the other hand, the data points are far from (1,1), the image is highly segmentable (see Fig. 4).

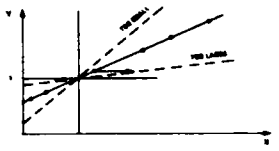


Fig. 4. Linearized pdf equation.

An error in α will not effect y, because $h^{WR}(J)$ and $h^{FR}(J)$ are obtained experimentally. The value of x will, however, be affected. If the calculated value of α (α_{cal}) is smaller than its correct value, α_c, the data point will appear closer to (1,1) and the performance measure will be more pessimistic than warranted. If we overestimate the value of α, $\alpha_{cal} > \alpha_c$, the data point appears further from (1,1). The performance measure will be overly optimistic (see Fig. 4).

The percent error in the value of $\left[1 - x(J)\right]$ can be derived by defining $\alpha_{cal} = \alpha_c + \alpha_E$

where α_{cal} = estimated value of α
α_c = correct value of α
α_E = the error in α

Also define $\left[1 - x\right]_{cal}$ = the value of (1-x) obtained by using α_{cal} and $(1-x)_c$ = the value (1-x) obtained by using α_c.

$$\text{The fractional error} = \frac{(1-x)_{cal} - (1-x)_c}{(1-x)_c} = \frac{\alpha_E}{1-\alpha_{cal}}$$

and is independent of the point (x,y).

The fractional error in x, i.e., $\frac{x_{cal} - x_c}{x_{cal}}$ is, however,

dependent of (x,y).

This discussion indicates strongly that the point (1,1) plays a unique role. To calculate the variance of the set of points $\{(x,y)\}$ about the point (1,1) use the following:

$$WSM = \sum_J h^{WR}(J) \left\{ \left[1 - y(J)\right]^2 + \left[1 - x(J)\right]^2 \right\} \tag{19}$$

and from equation 18 it can be seen that

$$(1-x) = \frac{1}{(1-\alpha)} \quad (1 - y) \tag{20}$$

and therefore,

$$WSM = \left\{ \frac{(1-\alpha)^2 + 1}{(1-\alpha)^2} \right\} \sum_J h^{WR}(J) \left[1 - y(J)\right]^2 \tag{21}$$

The usefulness of WSM as a performance measure arises in the fact that the percent error in WSM due to an error in α is independent of the distribution of data points (x,y). Thus, if we were to compare the performance of two features in segmenting the image, say intensity and edge magnitude, the ratio of WSM for the two features would be independent of the value of α.

Entropy

Several entropy measures have been tested. The entropy H is defined as:

$$H = -\sum_{J=1}^{N} P(J) \log_2 (P(J)) \tag{22}$$

where J = the probability of occurrence of feature value J and N = the number of feature values. H measures how evenly the feature values are occupied. If one feature value alone is occupied, then H = 0. If all feature values are evenly occupied, $H = H_{max} = \log_2 N$ (i.e., the number of bits allocated to the histogram). Obviously, for H = 0 an image is monotonic, but for $H = H_{max}$ the image contains the greatest possible variety. For example, if we count the number of unique adjacent pairs of intensity values an image contains, we would find just one unique pair for H = 0. However, we would find, on the average, the greatest number of unique pairs for an image when $H = H_{max}$. A small value of H indicates one or more narrow peaks in the feature value histograms, whereas a large value of H indicates a large deviation of the feature values.

The entropies for the window HW, target HT, and background histograms HB have been tested. In addition, $\Delta H = |HB - HT|$ has been used as a measure of the difference in the standard deviation of the background and target histograms. It is worth verifying if ΔH, combined with the difference in mean values between the background and target histograms, can parameterize the segmentability of the image. This appproach is taken in analogy with the difference in means, and ratio of the standard deviations, completely parameterizing two Gaussian distributions.

Computer simulations

The utility of the performance measures was tested using synthetic imagery. Two types of simulations were performed.

Image sequences

Twenty-five sequences, each containing 50 frames, of a Gaussian target moving through a a Gaussian background were generated on the HP-1000 and viewed on the I^2S. The target and background statistics are characterized by the equations

$$h^T(X) = \frac{1}{\sqrt{2\pi}\,\sigma_T} e^{-\frac{(X-m_T)^2}{2\sigma_T^2}} \text{ and } h^B(X) = \frac{1}{\sqrt{2\pi}\,\sigma_B} e^{-\frac{(X-m_B)}{2\sigma_B^2}} \qquad (23)$$

where X = grey level (it is transformed into the appropriate quantized value J)
m_T = the mean grey value of the target
σ_T = the standard deviation of the target grey values
m_B = the mean grey value of the background
σ_B = the standard deviation of the background values.

Two parameters were used to characterize the segmentability of the images. They are

$$\Delta = \frac{m_B - m_T}{\sigma_T} \quad \text{and} \quad \rho = \frac{\sigma_B}{\sigma_T} \qquad (24)$$

Fifteen sequences had unvarying statistics while 10 sequences had either Δ or ρ varying from frame-to-frame. The crudest segmentation procedures were implemented. α was set to 0.75 for all sequences. The maximum likelihood decision criteria was used, i.e., A = 2. No time series integration to minimize short-term statistical variation was attempted (see Table 1).

TABLE 1. CATALOGUE OF SYNTHETIC IMAGE SEQUENCES

Sequence Number	Mean Grey Level	No. of Grey Levels Spanned by σ_T	Δ	ρ	The Increment in Δ per Frame	The Increment in ρ per Frame
SEQ 606	80.0	10.0	3.0	2.0	N	N
SEQ 607	80.0	10.0	2.5	1.0	-0.1	N
SEQ 608	127.0	5.0	0.0	10.0	N	-0.198
SEQ 609	127.5	5.0	0.0	10.0	N	-0.3
SEQ 610	80	10	0	0.1	N	N
SEQ 611	80	10	0	0.5	N	N
SEQ 612	80	10	0	1.0	N	N
SEQ 613	80	10	0.5	0.1	N	N
SEQ 614	80	10	0.5	0.5	N	N
SEQ 615	80	10	0.5	1.0	N	N
SEQ 616	80	10	1.0	0.1	N	N
SEQ 617	80	10	1.0	0.5	N	N
SEQ 618	80	10	1.0	1.0	N	N
SEQ 619	80	10	2.0	0.1	N	N
SEQ 620	80	10	2.0	0.5	N	N
SEQ 621	80	10	2.0	1.0	N	N
SEQ 622	80	10	2.5	0.1	-0.1	N
SEQ 623	80	10	2.5	0.5	-0.1	N
SEQ 624	80	10	2.5	1.0	-0.1	N
SEQ 625	80	10	0	2.0	N	-0.0416667
SEQ 626	80	10	0.5	2.0	N	-0.0416667
SEQ 627	80	10	1.0	2.0	N	-0.0416667
SEQ 628	80	10	2.0	2.0	N	-0.0416667
SEQ 629	80	40	0.0	0.1	N	N
SEQ 630	80	10	0.0	10.0	N	N

Unitary frames

Single frames were generated on the VAX-780. The target and background statistics were again Gaussian, but the more general minimum risk rule was applied, i.e., A varied. In addition, frames which represented the whole range of α values were examined. Analytical results can be obtained for Gaussian distributions and these are compared with the results of the simulation. The analytical results can be thought of as the limiting case when the number of pixels in the image approach infinity. For example, the performance measure cos θ can be expressed analytically in the following manner:

$$\cos\theta = \left[\frac{2\rho}{1+\rho^2}\right]^{\frac{1}{2}} e^{-\Delta^2/2(1+\rho^2)} \tag{25}$$

Also, the locations of the decision criteria J_C can be analytically determined. The following two cases apply:

1. $\sigma_T = \sigma_B$

$$J_C = \frac{\Delta}{2} + \frac{1}{\Delta} \ln\left\{\frac{(1-\alpha)\,C\,(B/T)}{\alpha C\,(T/B)}\right\}$$

where $\Delta = \frac{m_B - m_T}{\sigma T}$, the distance between the two means measured in standard deviations.
As $\alpha \to 0$, $J_C \to +\infty$ and as $\alpha \to 1$, $J_C \to -\infty$

2. $\sigma_B = \rho\sigma_T, \rho \neq 1$

$$J_C = \frac{-\Delta \pm \left[\rho^2\Delta^2 + 2(\rho^2-1)\,\rho^2 \ln\left\{\frac{\rho(1-\alpha)C(B/T)}{\alpha C(T/B)}\right\}\right]^{\frac{1}{2}}}{\rho^2-1}$$

The two values of J_C will be J_C^1 and J_C^2

where $J_C^1 = \frac{-\Delta + [\quad]^{\frac{1}{2}}}{\rho^2-1}$,

$J_C^2 = \frac{-\Delta - [\quad]^{\frac{1}{2}}}{\rho^2-1}$

J_C can have zero, one, or two solutions. Suppose $\rho > 1$. As $\alpha \to 0$ $J_C^1 \to +\infty$ and $J_C^2 \to +\infty$, that is all $\{J\}$ are to be labeled "target." As α becomes larger, the two values of J_C approach $\frac{-\Delta}{\rho^2-1}$ and the set $\{J_T\}$ narrows, $J_C^1 < J_T < J_C^2$. When α reaches a critical value there is one solution, $J_C^1 = J_C^2 = \frac{-\Delta}{\rho^2-1}$. If α becomes larger still, there will be zero solutions and $\{J\}$ is labeled "background." Suppose $\rho < 1$. As $\alpha \to 0$ there are no solutions and all J are labeled "target." As α reaches a critical value there will be one solution and $J_C = \frac{-\Delta}{\rho^2-1}$. As α becomes larger still $J_C^1 < \{J_B\} < J_C^2$. When $\alpha \to 1$ all $\{J\}$ are labeled "background."
The HR and FAR can then be analytically determined using both the decision criteria calculated here and approximations to the Gaussian cumulative distribution function.

Results

The following types of questions have been addressed:

1. How well do the performance measures correlate with the statistics of the images? In our case the statistics are completely specified by Δ, the difference in the means, and ρ, the ratio of the variances of the target and background histograms.
2. How well does each calculated performance measure correlate with the actual performance of the algorithm? For example, how well does the calculated hit rate correlate with the actual hit rate?
3. How large is the range of values that a performance measure takes as a function of the variation in the statistics of the images?
4. Are any performance measures redundant? That is, are there pairs of performance measures which correlate well with each other?

Figures 5 through 9, obtained from the VAX-780 simulations, show the correlation of some of the performance measures with Δ and ρ. Figure 5 illustrates that the Gaussian statistics can be parameterized by Δ and ρ. The legend "infinite (∞) pixels" indicates the results were obtained analytically and are not subject to the statistical fluctuations which occur when a finite number of pixels are sampled.

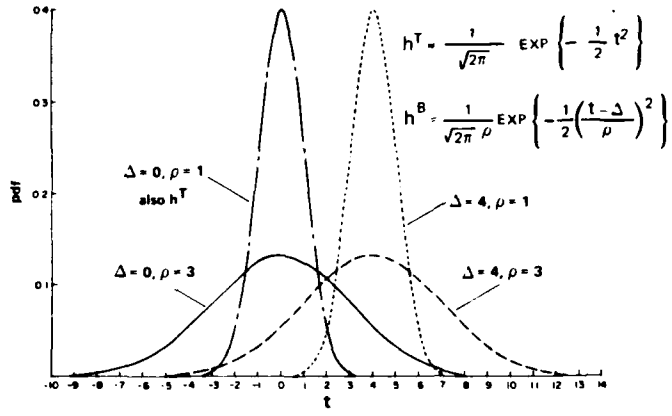


Fig. 5. Gaussian background distributions as a function of Δ and ρ.

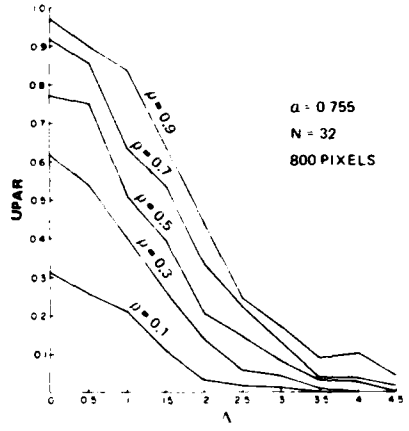


Fig. 6. Unreliability parameter vs. Δ.

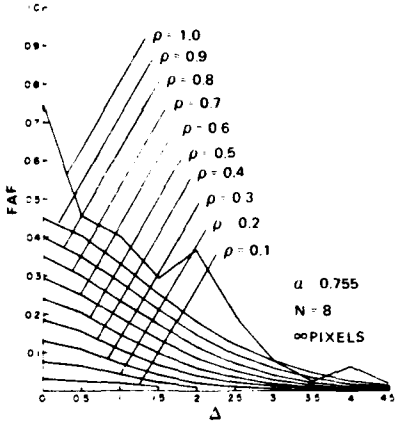


Fig. 7. False alarm fraction (FAF) vs. Δ.

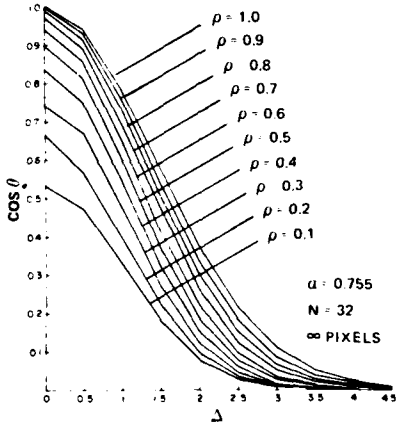


Fig. 8. Cos θ with ρ varying from 0.1 to 1.0.

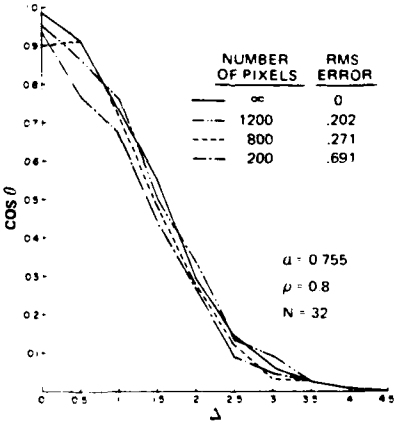


Fig. 9. Cos θ with a varying number of pixels.

To test the performance measures fairly for a finite number of pixels, the image sequences which were generated on the HP-1000 were subjected to two operations. The means and standard deviations of all performance measures were calculated. These were compared to the parameters Δ and ρ which characterize the segmentability of the images. Each performance measure was calculated in two different ways. One value was obtained by presuming omniscience. The target and background histograms were obtained by scanning the target and background domains in the window region. The performance measures obtained in this manner are the control results. The other value was obtained by assuming the same ignorance of the target and background histograms that were obtained for the tracker. $h^T(J)$ and $h^B(J)$ were obtained from $h^{FR}(J)$ and $h^{WR}(J)$ and are test results. The correlation function for the control and test results of each performance measure over the 50 frames of each sequence were calculated. These values are called the self-correlations. The results indicate that when the performance measures have a high standard deviation, such as those obtained for SEQ 622, the self-correlation is high (see Fig. 10 through 14). Note that the peaks in cos θ, UPAR and Bays coincide with each other and coincide with the valleys of HR. Note that Bays is less than 0.25 for the entire sequence because $\alpha = 0.75$. The maximum likelihood criterion makes it probable that under very poor segmentation conditions when $\Delta \to 0$ and $\rho \to 1$, all pixels will be labeled "background" if $\alpha > 0.5$. Thus, for poor segmentation conditions

$$\text{BAYS} \to (1 - \alpha) \text{ for } \alpha > 0.5 \text{ and BAYS} \to \alpha \text{ for } \alpha < 0.5$$

and Bays is therefore in general < 0.5.

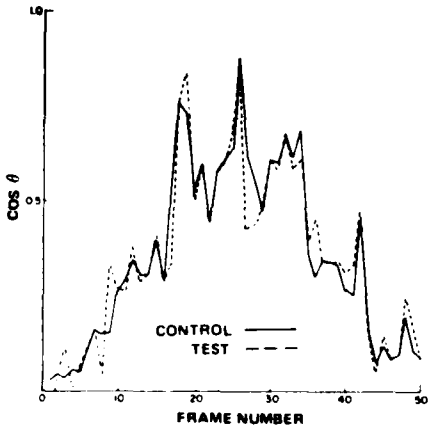


Fig. 10. SEQ 622, cos θ.

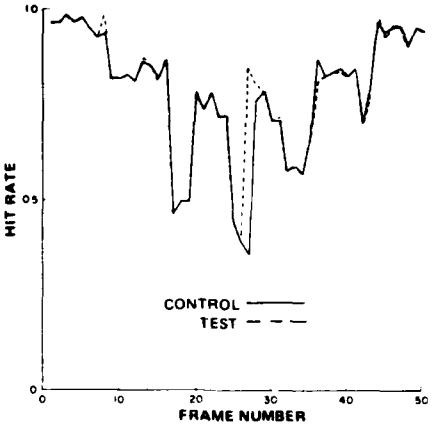


Fig. 11. SEQ 622, hit rate.

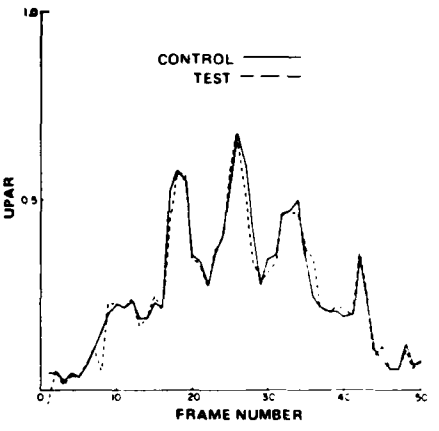


Fig. 12. SEQ 622, UPAR.

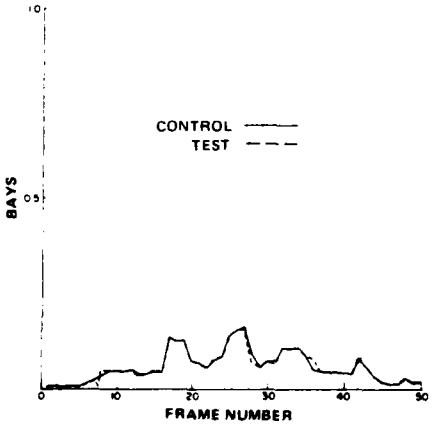


Fig. 13. SEQ 622, Bays.

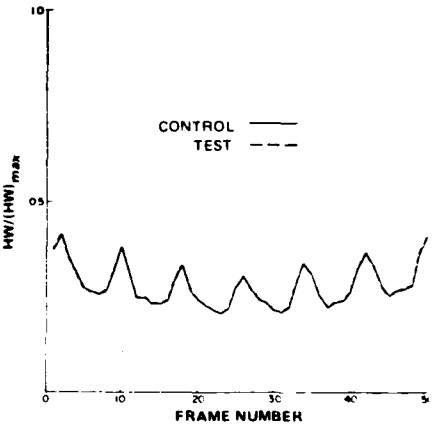


Fig. 14. SEQ 622, Entropy — $HW/(HW)_{max}$.

This argument implies that when $\alpha > 0.5$ the errors tend to be misses rather than false alarms and vice versa. This may explain the lack of variation in the FAR for most of the data.

One phenomemon that cannot be explained at present is the coincidence of the peaks of the entropy in Fig. 14 with the valleys of the HR in Fig. 11. The general trend of the graphs, as expected, is the same, but the peaks should also coincide.

In summary most of the performance measures appear to be useful. The WSM did poorly and needs to be dropped or reworked. Cos θ and UPAR are highly correlated and are therefore redundant. The FAR and FAF vary little due to $\alpha > 0.5$. Both performance measures need to be tested further on imagery for which $\alpha < 0.5$. For such imagery the HR is expected to deviate little. Performance measures such as FAR, FAF, and entropy, which have small standard deviations, can be useful as alarms that indicate a drastic variation in the tracking condition when they themselves change. Cos θ, Bays, HR, and UPAR appear to be the most sensitive performance measures.

Figure 15 through 20 are from SEQ 623. Note that the test performance measures vary more and on the average give a more optimistic estimate than the control. Also note the high correlation between cos θ and UPAR illustrated in Fig. 15.

The graphs from SEQ 628 (see Fig. 21 and 22) show that the performance measures are less sensitive to variations in ρ than to variations in λ. Although the self-correlation is poor (0.657 for Fig. 21), note that the standard deviations are also smaller. Also note that the self-correlations would be considerably improved if some type of averaging or integration were performed.

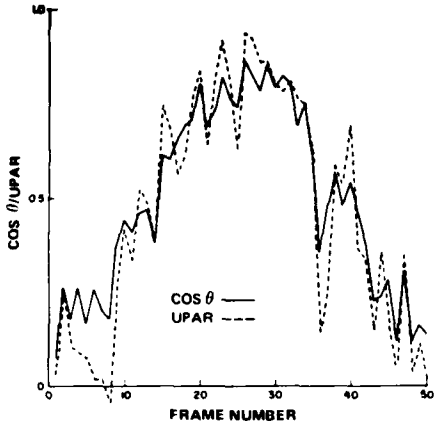


Fig. 15. SEQ 623, cos θ, UPAR; test results

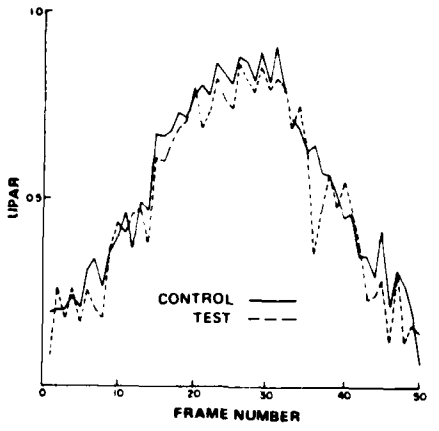


Fig. 16. SEQ 623, UPAR.

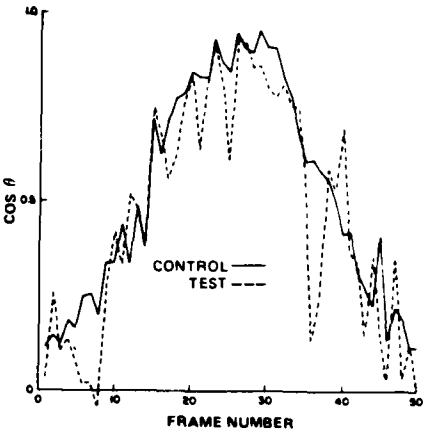


Fig. 17. SEQ 623, cos θ.

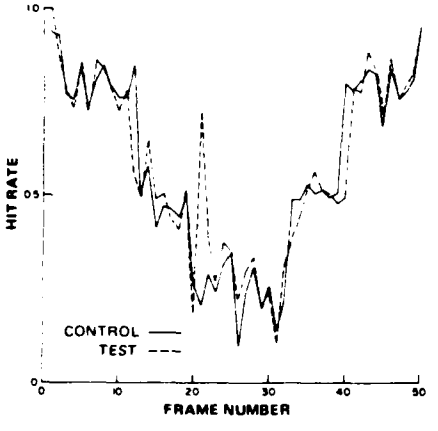


Fig. 18. SEQ 623, hit rate.

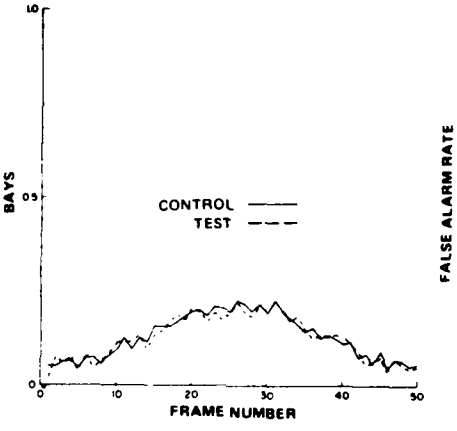


Fig. 19. SEQ 623, Bays.

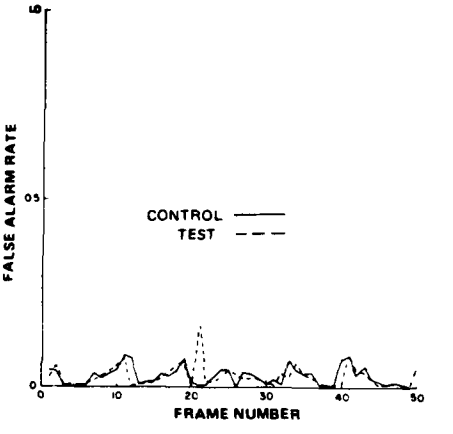


Fig. 20. False alarm rate.

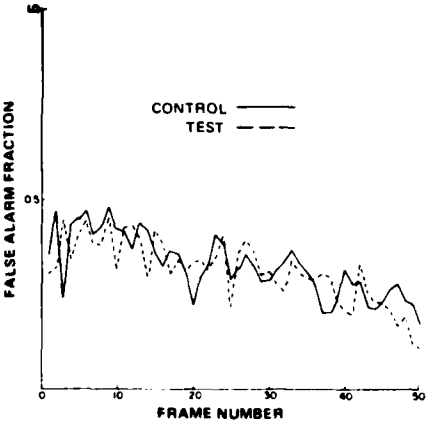


Fig. 21. SEQ 628, false alarm fraction.

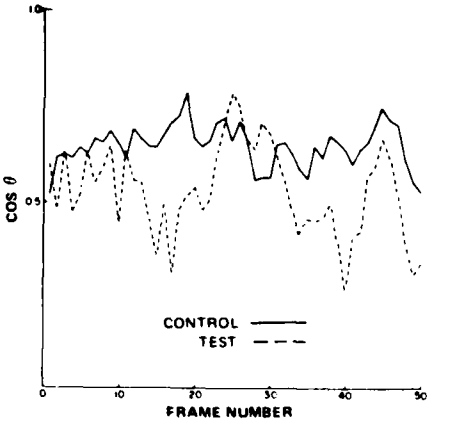


Fig. 22. SEQ 628, cos θ.

SEQ 610 and SEQ 629 have the same values for Δ and ρ. The difference between the sequences is that σ_T spans 10 grey levels for SEQ 610 and 40 grey values for SEQ 629. HW differs considerably for the two sequences as illustrated in Table 2.

TABLE 2. NUMBER OF BITS VS. ENTROPY OF WINDOW

Number of bits	SEQ 610 $HW/(HW)_{max}$	SEQ 629 $HW/(HW)_{max}$
2	0.05	0.31
3	0.07	0.34
4	0.29	0.50
5	0.32	0.50
6	0.34	0.56

A comparison of the other performance measures shows that SEQ 629 is more segmentable than SEQ 610. Table 3 illustrates the correlation of HW with Δ and Table 4 does the same for ΔH and ρ. Note the variation of HW with Δ in Fig. 23 and note the higher average values when compared to Fig. 14. Figure 24 illustrates the correlation between HW and HR.

TABLE 3. ENTROPY OF THE WINDOW HISTOGRAM VS. Δ.

Sequence Number	ρ	Δ	$\frac{HW}{(HW)_{max}}$
SEQ 110	.1	0.0	.32
SEQ 113	.1	0.5	.21
SEQ 116	.1	1.0	.25
SEQ 119	.1	2.0	.26
SEQ 111	0.5	0.0	.37
SEQ 114	0.5	0.5	.37
SEQ 117	0.5	1.0	.40
SEQ 120	0.5	2.0	.45
SEQ 112	1.0	0.0	.48
SEQ 115	1.0	0.5	.48
SEQ 118	1.0	1.0	.50
SEQ 121	1.0	2.0	.55

TABLE 4. ΔH VS. ρ

Sequence Number	Δ	ρ	ΔH* (maximum entropy = 5.0)		Standard Deviation of ΔH	
			Experimental	Control	Experimental	Control
SEQ 613	0.5	0.1	2.33	2.35	0.10	0.10
SEQ 614	0.5	0.5	0.73	0.86	0.17	0.10
SEQ 615	0.5	1.0	0.35	0.10	0.31	0.07
SEQ 616	1.0	0.1	2.19	2.10	0.10	0.11
SEQ 617	1.0	0.5	0.60	0.85	0.26	0.12
SEQ 618	1.0	1.0	0.34	0.10	0.29	0.07
SEQ 619	2.0	0.1	2.34	2.34	0.07	0.07
SEQ 620	2.0	0.5	0.53	0.85	0.30	0.10
SEQ 621	2.0	1.0	0.55	0.08	0.42	0.06

*Average difference between target and background entropies, H.

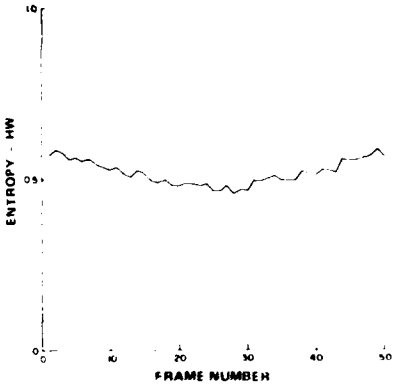


Fig. 23. SEQ 624, entropy-HW.

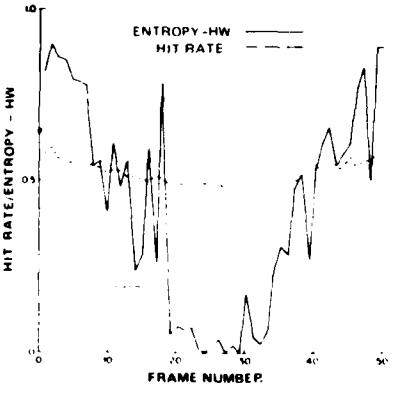


Fig. 24. SEQ 607, entropy — HW, hit rate.

Future research

Bimodal and trimodal synthetic Gaussian sequences will be generated and the performance measures will be tested. The performance measures will be integrated into MFBIT. The calculation of entropy is computationally costly and a replacement is being sought. A function of the grey level distribution of the histograms which is an extremum is needed when all grey levels are uniformly occupied. One possibility being considered is

$$SHIR \quad \pi_{J=1}^{N} h(J)$$

where $h(J)$ = number of pixels with grey level J with the proviso that $h(J') = 1$ if no pixels occupy grey level J'. Another possibility is to calculate the average absolute deviation of the grey value occupation from a uniform distribution.

Conclusion

It appears that the performance measures cos θ, UPAR, HR, FAR, and Bays are useful measures for statistical segmentation algorithms. Their validity and reliability have been proven by computer simulations. The entropy measures may be used to characterize the probability distribution functions (pdf) in a manner analogous to the characterization of Gaussian distributions by the mean and standard deviation. The entropy is less sensitive to variations in the pdf and can be used to signal drastic changes in the imagery. Some of the performance measures, such as cos θ and UPAR, are highly correlated with each other. It is therefore likely that only a subset of the performance measures will be implemented on MFBIT.

Acknowledgment

The work described in this paper has been supported in part by the Army Research Office under contract DAAG29-81-C-0034.

References

1. A.L. Gilbert et al., "A Real-Time Video Tracking System," Optical Engineering, vol. 18, no. 1, pp. 25-32, January/February 1979.
2. G.M. Flachs et al., "A Real-Time Video Tracking System," Annual report for contract DAAD-77-C-0046, Dept. of Electrical and Computer Engineering, New Mexico State University, NM, January 1979.
3. W.B. Schaming, "Adaptive Gate Multifeature Bayesian Statistical Tracker," submitted to SPIE 26th Annual International Symposium, San Diego, CA, August 1982.
4. J. Neyman and E.S. Pearson, "On the Use and Interpretation of Certain Test Criteria for Purposes of Statistical Inference," Biometrica, Part I: pp. 175-240, Part II: pp. 263-294, 1928.

END

FILMED

4-85

DTIC